juli-augustus '85

MSX CLUB magazine

NIEUW

- New Logo MSX
- 9 Grijstinten Screencopy
- Spritemaker
- Manic Miner
- Computer-adventure
- Coördinatenzoeker
- Soundmaster
- TRON-Info
- Videodisplay deel 2
- De Happy Hooker
- Cursus Z80 machinetaal deel 1
- Sony Plotter-printer deel 2
- Programmeertechnieken

een uitgave van MSX-club BELGIË - NEDERLAND

MSX-club

MSX-CLUB is een vereniging voor MSX-gebruikers in België en Nederland. Aansluiten bij de MSX-CLUB betekent dat U kan gebruik maken van de kennis en inzet van vele collega's MSX-ers. We organiseren regelmatig bijeenkomsten in België en Nederland waar ervaringen en ideeën kunnen uitgewisseld worden. Voor dringende problemen kan U contact opnemen met onze medewerkers.

MSX-magazine

In onze tweemaandelijkse uitgave vindt U vele programma's en bijdragen die U wegwijs maken in de complexe wereld van MSX-BASIC, machinetaal, hardware en de vele andere aspecten van het MSX-gebeuren. We melden U wat er nieuw is op de markt, publiceren testrapporten en brengen een overzicht van hetgeen U kan vinden in binnen- en buitenlandse tijdschriften. We hebben uitwisselingsakkoorden met verschillende buitenlandse uitgevers.
Uw bijdragen zijn belangrijk, het gaat tenslotte om UW MSX-computer. Uw vragen, antwoorden, programma's en artikelen zullen mede de richting en de inhoud van ons tijdschrift bepalen.

MSX-soft

Naast onze eigen uitgave bieden wij een geselecteerde keuze uit binnen- en buitenlandse pakketten.

Uw programma's kunnen ook opgenomen worden in onze bibliotheek, U krijgt een aantal programma's in ruil of U kan opteren voor een aangepaste financiële vergoeding.

We hopen dat U hier 3 keiharde redenen hebt gevonden om vandaag nog lid te worden van de MSX-club, kruip eens in de pen of achter het toetsenbord en laat wat van U horen.
Beschikt U over redelijk wat vrije tijd en bent U al behoorlijk thuis in de MSX-wereld dan kan U misschien als vaste medewerker onze redactie komen versterken. We spreken dan verder wel af hoe we uw inzet en onkosten kunnen vergoeden.

lidmaatschap

tarieven MSX-lidmaatschap + abonnement
(5 nummers in 1985 het eerste nummer verschijnt in april):
België: 650 Bfr. - Nederland: 36 Fl.
betalingswijze:
1. opsturen van Eurocheque naar: MSX-ledenadministratie p/a B. Van Rompaey, Bovenbosstraat 4, 3044 Haasrode, België (016) 46 10 85
2. overschrijving op bankrekening:
   voor België: Gen. Bankmaatschappij Leuven 230-0045353-74
   voor Nederland: AMRO-bank Baarle-Nassau: 46.07.36.051

Er bestaat ook de mogelijkheid om samen met het tijdschrift alle gepubliceerde programma's op cassette te ontvangen: prijs lidmaatschap + tijdschrift + cassetteabonnement: België 2.750 Bfr. - Nederland 150 Fl.



magazine

TWEEMAANDELIJKS TIJDSCHRIFT VOOR MSX-GEBRUIKERS

een uitgave van MSX-club, afdeling van DAInamic VZW Mottaart 20, B-3170 Herselt

redactie:
Dirk Bonné
Freddy De Raedt
René Rens
Bruno Van Rompaey
Jef Verwimp
Willy De Winter
Herman Bellekens
Frans Couwberghs
Guido Goyvaerts
Daniel Goyvaerts
Willy Coremans
Just Van Dunné

hoofdredacteur Nederland:
Frank Druijff (010) 25 42 75
secretariaat Nederland:
p/a Rinus Vijverberg
De Hauwklaver 6
3069 DJ ROTTERDAM

hoofdredacteur België:
Wilfried Hermans
(014) 54 59 74
secretariaat België:
p/a Mottaart 20
3170 Herselt

correspondent Frankrijk:
Cedric Dufour

correspondent England:
Dave Atherton

fotografie:
Paul Neuts

TRON-onderwijs rubriek:
Patrick Wynants

vormgeving:


advertentie-exploitatie:
Herman Bellekens

MSX club MAGAZINE

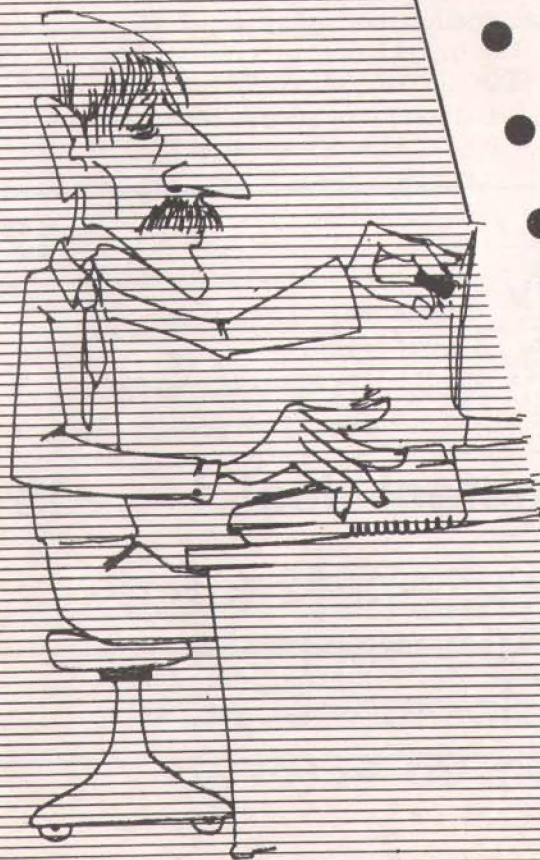
Beste Lezer,MSX-gebruiker,

Na een paar uitgaven van MSX-MOZAIK heeft de Belgische club een eigen gezicht gekregen met deze MSX club MAGAZINE. De samenwerking met de Nederlandse club blijft bestaan, U zal dan ook nog regelmatig bijdragen van onze noorderburen vinden in onze uitgave. In dit nummer vindt U vele programma-listings. Het is onze uitdrukkelijke bedoeling dat deze probleemloos kunnen ingetikt worden. We willen U de frustraties en ons de telefoontjes besparen. Waar we bij vorige gelegenheden onze DAISYWHEEL-printer gebruikten, hebben we nu onze MATRIX-printer een aantal nieuwe truukjes geleerd.(Lees hierover meer op p.33). De programma's worden gelist in een kolombreedte van 32 karakters.Indien de lijn langer is dan 32, worden de vervolglijnen afgedrukt met vooraan een aantal spaties. Dit aantal is bepaald door de lengte van het lijnnummer + 1. Woorden worden steeds gekapt als de kolombreedte bereikt is. REM-statements krijgen een duidelijke omlijning, dit komt de structuur van het programma ten goede. Onze printer kent nu alle tekens van de MSX-karakterset. Hoe U een bepaald teken vanaf het toetsenbord moet ingeven, vindt U terug in de tabellen op bladzijde 34 en 35. Binnen een string (tussen quotes) worden alle spaties vervangen door de kleine driehoek (CHR$(206)). ▲ Zo kan U gemakkelijk het juiste aantal spaties tellen en correct intikken. We stellen dan wel voor om dit teken (▲) voortaan niet meer in programma's te gebruiken. Indien U dit teken toch nodig heeft, kan U nog steeds CHR$(206) intikken.

We wensen U veel lering en vermaak met deze eerste uitgave. Mogen we de volgende keer ook uw bijdrage opnemen?

Met vriendelijke groeten, de redactie





Inhoudstafel

MSX

Programmeertechnieken 4

ON ERROR GOTO,
ERROR,
ERR,
ERL,
RESUME,
RESUME NEXT

In zijn vaste rubriek "programmeertechnieken" behandelt Frank Druijff een aantal onderwerpen waar een beginnende programmeur vast mee te maken heeft. **Als je bij uw eerste programma er al in slaagt een programma te schrijven dat volledig door de computer wordt begrepen, ook al zijn de resultaten niet wat U wilde, dan bent U op de goede weg een erg goede programmeur te worden.** Niet erg hoopgevend, maar Frank geeft wel de nodige informatie om een aantal klassieke addertjes te vermijden. Meteen behandelt hij ook de plus- en minpunten van MSX-BASIC. Naast SPOORZOEKEN in BASIC-programma's worden ook een aantal editeer-tips gegeven. We zijn er van overtuigd dat U al uitkijkt naar de volgende aflevering...



De Happy Hooker 8

U mag niet verwachten hier een bijdrage van Xaviera te vinden. Aan de hand van een eenvoudig voorbeeld wordt het begrip **HOOK** behandeld. Als U dit verhaal heeft doorgenomen kan U met een paar **POKES** BASIC verplichten, de informatie die hij naar het scherm stuurt ook naar de printer te sturen zonder uw programma te moeten wijzigen. Andere toepassingen op de HOOK-structuur volgen...

9 Grijstinten Screencopy 10



Het **SCREENCOPY PROGRAMMA** is een van de eerste realisaties van ons team programmeurs. Met deze routine kan U de scherminhoud van een van de MSX-screenmodes haarfijn op papier zetten.De schermkleuren worden weergegeven in 9 goed-onderscheiden grijstinten. Ook bruikbaar in combinatie met grafische programma's zoals EDDY II of GREATIVE GRAPHICS. Een primeur,..een must indien U over een grafische printer beschikt.

Coördinatenzoeker 12



Kaart-coordinaten ingeven wordt kinderspel met het programma **COORDINATENZOEKER** van R.Ceuterick. Waarschijn-lijk vindt U nog wel andere toepassingen voor dit programma.

Cursus Z80 machinetaal deel 1 15

Cursus Z80

Bruno Van Rompaey en Jaak Snauwaert zijn uw reisgidsen op verkenningstocht naar de meest intieme plaats van uw MSX-computer **: de microprocessor** .De eerste afleveringen zullen ons de nodige voorkennis verschaffen om later effectief te kunnen programmeren in machinetaal. Het begin van een lang verhaal...

club MAGAZINE

20 Sony Plotter-printer

In het tweede deel van het **SONY PLOTTER/PRINTER** verhaal komen verdere mogelijkheden in tekstmode aan bod. In een volgende aflevering gaan we met dit veelzijdige kleinood op de grafische toer.

28 Soundmaster

Met de SOUND-commando's bedienen we rechtstreeks de registers van de audio-chip in onze MSX. **SOUNDMASTER** laat U op een eenvoudige manier deze register bedienen. Noteer de gegevens als U een interessante klank gevonden hebt, dit kan je later gebruiken in uw eigen programma's.

30 Computer-adventure

Wim Dewijngaert heeft een bestaand **computer-adventure** grondig gewijzigd . Aan de hand van de Nederlandse reacties van de computer moet U proberen uw zoektocht naar computer-materiaal te voltooien. Goede reis !

33 MSX-TRANS

de routine die uw printer omtovert tot een volwaardige MSX -printer met vele EXTRA MOGELIJKHEDEN ..

36 Spritemaker

R.CEUTERICK biedt ons een **SPRITEMAKER** met klasse. Het minder prettige tikwerk wordt vermeden door de SPRITE-gegevens op cassette of schijf te bewaren. De schermplaatjes op p.38 zijn ook gerealiseerd met ons screen-copy programma.

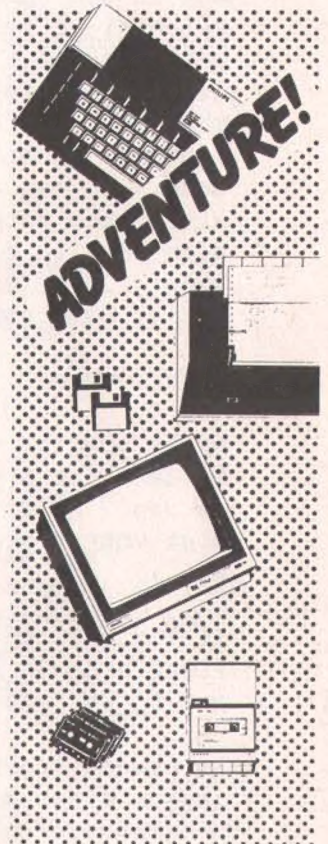
42 TRON-Info

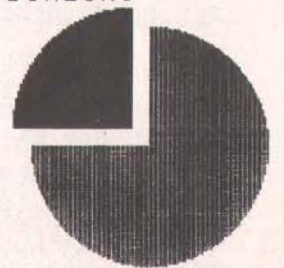
Het **TRON-team** heeft voor ons 3 programma's uit de educatieve sfeer getest : NOTENLEER, LESSEN IN BASIC EN LOGO. Hun bevindingen op p.42 en 43.

44 Videodisplay deel 2

Frans Couwberghs brengt de tweede aflevering van **VIDEODISPLAY.** Deze keer worden SCREEN 2 en de verschillende grafische opdrachten behandeld.







(c) MSX-club Belgium
Mottaart 20
3170 Herselt
(014)54 59 74

Programmeertechnieken

Inleiding

In deze tweede aflevering van de rubriek programmeer-technieken wil ik beginnen om met U programmeerfouten te bespreken.
Dit zal vermoedelijk minstens twee maar misschien nog meer artikelen beslaan. Ik betreur dat voor de mensen die graag alles netjes bij elkaar hebben. Ik verwacht echter wel reacties van andere gebruikers, die mij verder kunnen helpen. Meent U dus dat ik een voor dit onderwerp belangrijke zaak over het hoofd heb gezien schrijf of bel mij gerust.
Gezien de huiselijke omstandigheden moet ik U echter wel verzoeken bij voorkeur op dinsdagavond te bellen.
Schrijven mag ook maar verwacht niet altijd een antwoord. Ik lees echt alle post en zal de aangedragen tips, aanwijzingen en wat dies meer zij altijd gaan gebruiken in deze serie.

Fouten

Welke soorten fouten kunnen er tijdens het programmeren gemaakt worden ? Nu dat zijn er nogal wat. Niet voor niets zeggen de amerikanen wel :

If the only thing you get from your first program are wrong answers, you are to be congratulated.

In gewoon nederlands wil dit zoveel zeggen als : Wie bij zijn eerste programma er al in slaagt een programma te schrijven dat volledig door de computer wordt begrepen, ook al zijn de resultaten niet wat U wilde, dan bent U op de goede weg een erg goede programmeur te worden.

Velen zullen al in de uurtjes die zij achter het toetsenbord zaten, gemerkt hebben dat het een zeldzaamheid is als een programma ook de hele kleine ineens goed lopen. En als ze dan goed lopen blijken de antwoorden die we krijgen vaak weer niet die antwoorden te zijn die wij nu net met dit programma wilden verkrijgen.

Hoe onstaan nu deze fouten ?
Hoe kunnen we ze voorkomen ?
En meest belangrijke : hoe komen we er achter wat er fout is en waar die fout staat ?

MSX-basic

De MSX-basic is een enigzins hybride vorm van gebruikersvriendelijkheid en gebruikersonvriendelijkheid. Wie ooit op een Sinclair of zoals ik op een DAI heeft gewerkt zal weten dat er veel betere mogelijkheden van foutcontrole en foutinterceptie zijn, maar aan de andere kant, wie ooit met een apple heeft gewerkt of zelfs het gewone Microsoftbasic, zal de MSX-basic een verademing vinden.

Wat is er goed en wat is er zwak in de MSX-basic zodat we daar rekening mee kunnen houden ?

Pluspunten

Goed is dat we fouten tijdens de programmaverwerking kunnen ondervangen daar zelf actie op kunnen nemen en het programma laten vervolgen zoals wij dat wensen. Al gelden ten aanzien van dat laatste wel enige restricties, waar ik later op zal terugkomen.
Goed is ook de mogelijkheid om in de meeste gevallen te weten welk soort fout werd gemaakt en waar d.w.z. in welke regel die fout gemaakt werd.
Hier hebben dus al gelijk een tip :

****

T I P

Zet in elke regel maar een instructie neer. Loopt het programma dan stuk dan kunnen we opvragen, waar het stuk liep. Weten we dan de regel, dan weten we ook gelijk de instructie die de fout veroorzaakte.

****

Ook is goed de mogelijkheid ondanks een vastgestelde fout een programma te vervolgen, maar we moeten wel uitkijken met het gebruik van deze

faciliteit daar het snel tot programma's leidt die alsmaaar doorgaan en allang niet meer met iets zinnigs bezig zijn.
Eigenlijk al genoemd maar toch belangrijk genoeg om weer genoemd te worden vanwege een toepassing, die erg handig kan zijn. Ik doel hier op de mogelijkheid de fouten te ondervangen en dan de foutmelding in het nederlands op het scherm te zetten. Vooral in het onderwijs en bij gebruik van een programma door niet-deskundigen of mensen die het engels (nog) niet machtig zijn.
De eenvoudigste methode om programma's te beveiligen tegen stukgaan bij b.v. leerlingen is wel de oplossing om bij elke fout in het programma de tekst 'ROEP JE LERAAR' op het scherm te laten zetten.
Tot slot is foutmelding die we normaal krijgen een sterk punt in het voordeel van MSX-basic.

Minpunten

Wat kan er nog slecht zijn zult U misschien denken ? Nu, de interpreter van de basic tekst begint pas te werken als we het programma laten lopen. Beter zou zijn dat elke ingetikte regel direct na het indrukken van de return (enter) werd gecontroleerd op een aantal aspecten, die reeds in dat stadium te herkennen zijn als fouten tegen de syntax.
Een fout niet van de MSX maar van de programmeur, die een voordeel van MSX-basic niet juist gebruikt, is het feit dat een programma na een fout toch door kan gaan.
Een nadeel kan ook zijn dat we zonder het ons beseffen gedeeltelijk nog in een programma zitten. Ik zal een en ander nog met voorbeelden toelichten maar zeg nu al dat er verschillende mogelijkheden van reageren van de MSX-machine zijn zonder dat het ons duidelijk hoeft te zijn waarom dit zo is.

Instructies

MSX-basic kent een aantal instructies voor de hiervoor genoemde faciliteiten.
Ik verdeel deze in de instructies die direct met de foutenbehandeling te maken hebben en de instucties die de programmeur ten dienste staan om de goede werking van programma te controleren.

In de eerste groep onderscheiden we:

```
        ON ERROR GOTO,
           ERROR,
           ERR,
           ERL,
           RESUME,
en      RESUME NEXT
```

In de tweede groep onderscheiden we ten eerste een tweetal instructies, die wel niet direct met de bovenstaande instructies samenhangen maar wel ten nauwste zijn verbonden met het opsporen van fouten namelijk :

```
           TRON
en         TROFF.
```

Verder zien we in deze groep :

```
ON INTERVAL GOSUB
ON KEY GOSUB
ON STRIG GOSUB
```

En eventueel nog zaken als RENUM, LIST, BEEP, PEEK en dergelijke.

De laatste, hoewel zij weleens gebruikt worden bij foutdetectie zijn geen instructies die met het oog op deze foutdetectie ontworpen zijn.
Daar de TRON en TROFF het simpelste zijn te bespreken wil ik daarmee beginnen.

Spoorzoeken

TRON en TROFF betekenen respectievelijk TRace ON en TRace OFF.

Een trace is een spoor en dat is precies wat we hiermee kunnen doen. We verplichten de MSX-computer met de instructie TRON vanaf het moment dat deze instructie gegeven werd een spoor achter te laten.
Hij zal dan namelijk alle regelnummers geven die uitgevoerd worden. De tip die gegeven werd blijkt ook nu weer zijn vruchten af te werpen. Met slechts een instructie per regelnummer weten we zelfs precies met welke instructie hij bezig is.

Met de TROFF instructie kunnen we deze spoorplaatser weer uitzetten. Controleer nu eens zelf of de TROFF zelf nog wordt ge'trace'd.
Het spoor dat de TRON achterlaat is echter niet zo uitgebreid als we ons soms zouden wensen. Alleen het regelnummer wordt gegeven en dat is toch echt niet zoveel informatie.
Daarom is het handig het gedeelte van het programma dat we willen 'trace'n eerst even op papier te zetten.
Ook gaat het soms zo snel en schuift de tekst die we nodig hebben zo snel uit beeld dat we er nog geen nuttige informatie uit kunnen halen.
Het zal dus vaak nodig zijn het programma in kwestie te vertragen.
Dit kunnen we op meerdere manieren doen maar een methode ligt wel erg voor de hand :

Met een ON INTERVAL=xx GOSUB rrr met xx een geschikte kleine waarde en rrr het regelnummer waar de subroutine staat waar we bijvoorbeeld om een speciale toets vragen om door te mogen gaan geven we onszelf regelmatig voldoende tijd om de nodige gegevens over te nemen.

Deze onderbreking wordt dan vlak voor de TRON met INTERVAL ON geactiveerd.
Alle drie instructies kunnen best op dezelfde regel staan, omdat zij alle drie in de definitieve versie van het programma niet meer aanwezig zullen zijn.
Denk er wel aan in de subroutine steeds weer (wel aan het eind !!!) een INTERVAL ON te geven.

Een instructie per regel

Ik heb nu al een paar maal geschreven dat het handig is om maar een instructie per regel te hebben.
Dit is ongetwijfeld waar in het verband van foutendetectie maar bij programma's waar we absolute topsnelheid eisen is dit zeer beslist niet aan te raden.
Ook de overzichtelijkheid kan gebaat zijn bij meerdere instructies op een regel.Zie bijvoorbeeld :

```
70 MH=400:REM MH is de Maximum Hoogte
              of
70 FOR I=0 TO 200:NEXT:'Wacht even
```

Hoe kunnen we nu aan dit dilemma ontkomen ? Wel voor beginnende programmeurs is het recept simpel : de foutgevoelige stukken en dat is in het begin vrijwel alles worden ingetikt met een instructie per regel. Is dan alles in orde bevonden dan kunnen we de regels, die samengevoegd kunnen worden tot een regel gaan samenvoegen.

Samenvoegen

Neem als voorbeeld :

```
10 A=1
20 FOR I=1 TO 20
30 A=A*I
40 PRINT A
50 NEXT
60 END
```

En we willen nadat we gezien hebben dat dit goed werkt regel 20,30,40 en 50 tot een nieuwe regel combineren.
We maken een list van het programma en zoeken met de cursor de plaats direct achter de (tweede) 20 in regel twintig. We drukken daarna op b.v. de spatiebalk totdat de cursor op de (eerste) A van regel dertig staat.
Dan drukken we op de BackSpace toets tot de cursor weer direct achter de (tweede) 20 van regel twintig staat.
De A staat daar dan ook. Tot slot zetten we er met een insert ':' een dubbele punt tussen en de zaak lijkt gefikst. Geef nu een list en U ziet het laatste probleem : de oorspronkelijke regel dertig moet nog verwijderd worden. We doen dit met : 30 en dan direct return. (de toets natuurlijk en niet het woord)

De gevorderde, maar ook de zichzelf overschattende, programmeur zal meerdere instructies op een regel zetten en dan toch weleens moeten vaststellen, dat het niet allemaal zo loopt als hij zich gewenst had.
We hebben dus nu behoefte een regel met meerdere instructies om te zetten in een aantal regels met alle slechts een instructie.

Uitsplitsen

Neem als voorbeeld het volgende kleine programma. Bekijk het goed voordat U verder leest.

```
10 A=1
20 FOR I=0 TO 20:A=A*I:NEXT:PRINT A
30 END
```

Ziet U het subtiele verschil met het vorige programma ? Ja, hoor ik velen al zeggen de PRINT A staat nu na de NEXT en bij het vorige programma er voor. De programmeur was dus niet geinterresseerd in tussenresultaten maar alleen in het eindresultaat.

Jammer genoeg voor onze programmeur is dit niet het resultaat van dit programma. Tik maar in als U de fout niet ziet. Onze programmeur komt er niet uit en wil nu regel over vier regels verdelen. Ik geef U een simpel recept :

Geef een list. Ga dan met de cursor op de (eerste) 0 van regel 20 staan dus de nul van het regelnummer. Zet er een 1 neer en geef return. Zet de cursor op de 1 en zet er een 2 neer en geef weer return. En dan nog een keer met een 3 op de 2 neerzetten. Dan geven we een list en ons programma heeft nu vier identieke regels met de regelnummers 20,21,22 en 23. Verwijder nu met de toets del uit regel 20 de laatste drie instructies, uit regel 21 de eerste en en de laatste twee en analoog behandelen we regel 22 en 23. Het resultaat is dat regel 20 over de vier regels 20,21,22 en 23 is uitgesplitst zodat we bij de speurtocht waar nu de fout zit dit simpeler kunnen vaststellen.

Opgave

De hiervoor genoemde methodes om twee of meer regels tot een samen te voegen, respectievelijk om een regel over twee of meer regels uit te splitsen zijn zeer beslist niet de slimste methodes, maar ze werken wel en zijn simpel te onthouden.

Wat wij nu graag zouden willen is dat de lezers van dit artikel zelf eens gaan nadenken over methodes om het samenvoegen respektievelijk het splitsen beter te doen. Een mogelijkheid tot verbeteren heb ikzelf al gevonden met gebruikmaking van de CRTL (controltoets), maar de methodes blijven in principe gelijk.

Weet U een andere en natuurlijk betere methode dan willen wij U vragen deze methode in te sturen naar het volgende adres :

MSX-Club p/a F.H Druijff
's-Gravendijkwal 5a
3021 EA Rotterdam
tel. 010-254275

In het volgende nummer zal ik dan de beste inzendingen bespreken. Beter is misschien nog dat ik uzelf aan het woord laat om een en ander uit de doeken te doen. Dus als U er zelf al een artikel(tje) van wilt maken; graag ! Weet U wel iets, maar kunt U dat niet gemakkelijk onder woorden brengen stuur dan de methode in zodanig dat ik het in de volgende rubriek kan verwerken.

Nu de overige instructies uit de tweede groep. U zag reeds dat het alle instructies van het type ON .... GOSUB waren.

Ik wil met deze instucties vanzelfsprekend het normale programmaverloop onderbreken. In de subroutine waardan naar toegegaan wordt kan ik bijvoorbeeld de waarden van bepaalde variabelen controleren. Zijn die waarden anders dan door mij verwacht, zou het natuurlijk best eens zo kunnen zijn dat daarmee de fout gevonden is.

Tevens kunnen we met deze instucties een belangrijk nadeel van de trace ondervangen. De trace werkt namelijk normaal alleen in tekstscherm. Hebben we dus een grafisch programma dat ergens de fout in gaat, kunnen we dat niet met behulp van TRON nalopen. Wel kunnen we echter met een van de genoemde onderbrekingen een soort TRON ontwerpen, die geen uitvoer op het scherm geeft maar bijvoorbeeld op de printer. Of via de luidspreker vandaar de BEEP bij de genoemde instucties.

De volgende keer zal ik vervolgen met de instucties uit de eerste groep. Nu is daar geen plaats meer voor maar U houdt ze tegoed.

Frank H. Druijff

De Happy Hooker



DE HAPPY HOOKER ...

In de MSX-constructie bieden HOOKS een redelijk eenvoudige manier om de BASIC interpreter op bepaalde punten te onderscheppen. Dit opent de mogelijkheid om extra of andere activiteiten te laten uitvoeren, zelfs om bepaalde activiteiten van BASIC volledig te negeren. In de meeste uitgebreide handleidingen die ook iets willen vertellen over machinetaal programmeren. vinden we de complete lijst van deze HOOKS. Ze bevinden zich in het Ram-gebied &HFEOO-&HFFFF.
De meeste ROM-routines gaan helemaal in het begin van hun activiteit even "kijken" naar hun HOOK-adres .Dit gebeurt door het uitvoeren van een CALL, nadat de registers netjes op STACK bewaard zijn. Meestal vinden ze daar een RET(urn)-instructie, die ze meteen verplicht hun opdracht netjes uit te voeren. Het scenario wordt echter gewijzigd als we zelf de RETURN-instructie (liefst zinvol) vervangen hebben door een andere opdracht.

Aan de hand van een eenvoudig voorbeeld zullem we proberen aan te tonen dat deze HOOKS ook nuttig gebruikt kunnen worden zonder onmiddellijk af te dwalen in ingewikkelde machinetaal-toestanden.

Het probleem

Stel dat je een programma ontwikkeld hebt in de tijd toen je nog niet over een printer beschikte. Uiteraard worden alle resultaten netjes met PRINT op het scherm gebracht. Nu de nieuwe printer pas uit de doos is, willen we toch wel graag deze resultaten op papier bekijken.

Zit er dan echt niets anders op dan het hele programma te gaan editeren en alle PRINT'S te vervangen door LPRINT'S ? Uiteraard waren we niet aan dit verhaal begonnen als er geen oplossing in ons achterhoofd zat ...

Op zoek ....

In ons handboek (in dit geval de Complete Programmers Guide van Melborne) zoeken we de HOOK van de routine CHPUT. We weten immers dat de informatie naar het scherm gaat, we veronderstellen dan maar even dat HCHPUT hiervoor verantwoordelijk zal zijn.
Het handboek geeft hiervoor het adres:

&HFDA4

Met deze wetenschap kunnen we nu aan de CHPUT-hook gaan sleutelen zodat de informatie ook naar de printer gaat. In de lijst van Entry Points (plaatsen in de ROM waar BASIC naar binnen gaat om een taak uit te voeren) vinden we &HA5 = LPTOUT .(This CALL is used by the BASIC interpreter to write a character to the printer). Duidelijker kan het toch niet.
Vooraleer we met onze POKES in het rond beginnen te slingeren, zetten we onze bevindingen (en veronderstellingen) nog even op een rijtje:
HCHPUT = &HFDA4 : hier komt BASIC dus telkens langs bij het begin van zin CHPUT- routine.
&HA5= LPTOUT : deze routine zendt BASIC een character naar de printer. Dat dit gebeurt met de ASCII-waarde in de accumulator is zeer gebruikelijk, maar dit komt voor ons voorlopig niet aan bod. Om onze beeldspraak van daarstaks verder te zetten, zouden we kunnn zeggen dat we nu de deuren kennen waarlangs BASIC naar binnen gaat. We weten niet precies wat hij (zij?) daarbinnen doet,

maar we weten wel wat er gebeurd is als hij buiten gaat.
Het plan is dus als volgt opgezet : we zetten de HOOK (klinkt gezellig,niet?) op HCHPUT richting LPTOUT, in de hoop dat BASIC telkens hij een character naar het scherm stuurt, dit ook door zijn printerpoort stuurt.

De oplossing

Vanaf &HFDA4 moeten we een paar Z80 instructies plaatsen die een CALL naar &HA5 veroorzaken, gevolgd door een RETURN. Deze RETURN geeft de situatie terug in handen van de oproeproutine, die zijn opdracht dan maar verder getrouw moet uitvoeren.

Deze 4 POKES zouden het gewenste resultaat moeten geven:

```
POKE &HFDA4,&HCD (CALL)
POKE &HFDA5,&HA5 (LOW BYTE)
POKE &HFDA6,0    (HIGH BYTE)
POKE &HFDA7,&HC9 (RETURN)
```

Voor alle veiligheid nemen we deze commando's op in een paar lijnnummers , zodat we ons tikwerk nog eens goed kunnen nakijken. Dit is beslist nodig, een tikfout zou hier onvoorspelbare gevolgen kunnen hebben!

Als de printer nu goed aangesloten is, en hij staat aan, dan moeten alle boodschappen die op het scherm verschijnen ook op ons papier terechtkomen.(Inclusief alle ongewenste fout- en andere meldingen van BASIC.

Zo, dit resultaat tonen we tevreden aan onze huisgenoten.

We blijven de computer de baas ...

We willen dit verhaal graag afronden met een laatste POKE-instructie. Het zou namelijk netjes zijn als we de printer af en toe even konden buiten spel zetten, liefst zonder de AAN/UIT schakelaar te bedienen.
Dit kunnen we bereiken door op het eerste adres van onze ondertussen vertrouwde HOOK opnieuw een RETURN-opdracht te plaatsen en zo de originele toestand te herstellen.
Hier komt hij :

```
POKE &HFDA4,&hC9
```

Na deze POKE gebruikt BASIC terug de oude weg . De andere waarden kunnen op HOOK-adressen blijven zitten. Met een POKE (POKE &FDA4,&HCD) is onze extra routine immers opnieuw geactiveerd.

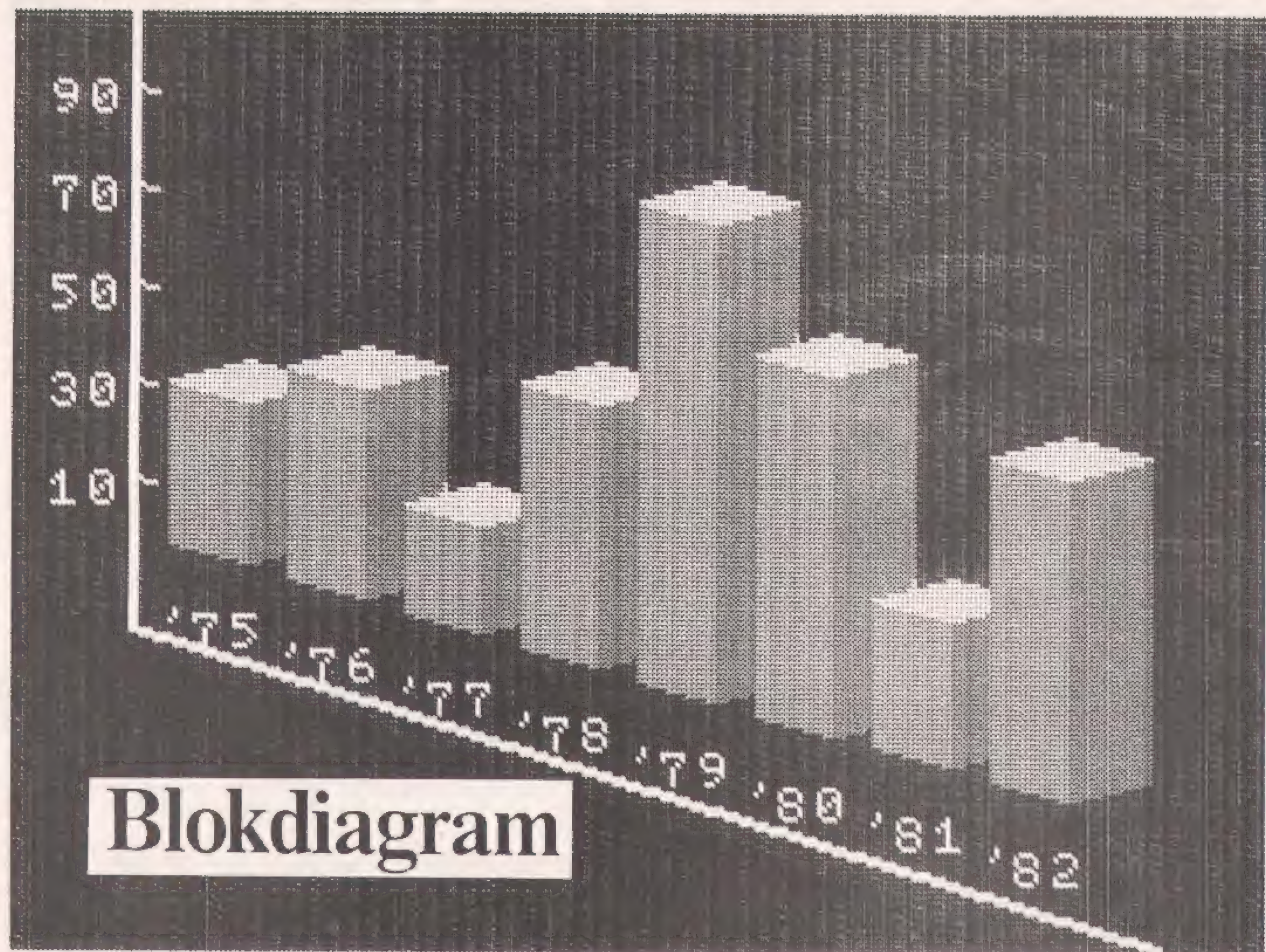
U zal waarschijnlijk al opgemerkt hebben dat de printer de gekregen informatie niet onmiddellijk afdrukt.De meeste printers hebben een buffer waarin de informatie wordt opgeslagen tot een CHR$(13) wordt doorgestuurd .

We hopen dat deze HAPPY-HOOKer , ondanks zijn misleidende titel U enige HOOK-inzichten heeft bijgebracht...

W.Hermans

```
10 REM
20 REM Schoon lieveken (c) BY WDW
30 REM
```

```
40 SCREENØ:WIDTH4Ø:COLOR15,14,14:CLS
50 KEYOFF:LOCATE13,11:PRINT"Schoon lieve
   ken"
60 A$="V15T1ØØO4L8DGG16GGB.G16GGAAEED4"
70 B$="S1M25ØØT1ØØO4L8DGG16GGB.G16GGAAEE
   D4"
80 C$="T155L8O5DDO4A4O5DDD4C#4D4DDD.E16D
   CO4BR8O5D"
90 D$="S1M25ØØT155L8O4DEF#4EFG4A4D4D8R8R
   4R8D8G.A16GF"
100 E$="T155L8O5E.D16CO4BA4R8O5DD.E16CO4
    AGABO5CO4B4AAG4G"
110 F$="S1M25ØØT155L8O4E4R8GA.G16FED4D4E
    4DCDGEFG4GR8"
120 PLAYA$,B$:PLAYC$,D$:PLAYE$,F$
130 IFPLAY(1)=-1THENGOTO14Ø
140 IFPLAY(2)=-1THENGOTO13Ø
150 PLAY"V14","V12"
160 A$="T155O4L8DGGGBO5D4O4BO5DE.D16CO4B
    A4ADG.G16GG"
170 B$="T155O4L8DGGGBA4GAB4G4E4EEG8.G16G
    G"
180 C$="T155O4L8B.G16GGAAEED4ABO5D4O4BO5
    DE4E4D.E16DCO4B4R8A"
190 D$="T155O4L8B.G16GGAAEED4FGA4GAB4A4F
    4R8AB4FE"
200 E$="T155O4B.O5C16O4BAG4O5D4C.O4B16AG
    B4A4G4GABO5CD4D4DR8"
210 F$="T155O4D4R8DE4G4E4F4G4GFE4FEDEGAB
    4BR8"
220 PLAYA$,B$:PLAYC$,D$:PLAYE$,F$
```



kleur 0
kleur 1
kleur 2
kleur 3
kleur 4
kleur 5
kleur 6
kleur 7
kleur 8
kleur 9
kleur 10
kleur 11
kleur 12
kleur 13
kleur 14
10 kleur 15

multistrike

9 grijstinten

SCREENCOPY

Hiernaast vindt U een paar resultaten van schermafdrukken met onze **9-grijstinten multistrike screencopy** routine.
Wat behelst deze erg lange titel ?
screencopy : een routine die de scherminhoud van het hoge-resolutie scherm (256 × 192) op papier kan weergeven.
9-grijstinten : MSX kan 15 kleuren weergeven. Deze kleurinformatie kan omgezet worden in grijstinten met de variatie van het aantal aangeslagen naaldjes van een matrixdrukker. Daar we echter met een matrix van 4×4 zouden moeten werken om deze 15 kleuren weer te geven, is het uiteindelijke formaat te groot voor een klassieke A4 printer.
We houden het daarom bij 3×3 resolutie : hieruit komen 9 goed onderscheiden grijstinten naar voor. Elk van de 49152 schermpunten wordt dus weergegeven met een aantal dotjes van Ø tot 9. Ø betekent dat we absoluut wit verkrijgen, 9 dotjes geven volledig zwart.
De overige kleuren worden per twee gegroepeerd, zodat kleuren 2 en 3 identiek worden weergegeven, ook 4 en 5, 6 en 7 enz.
Kleur Ø is bij MSX een geval apart : deze kleur duidt aan dat de BORDER-kleur genomen wordt.
Hier houdt de routine uiteraard ook rekening mee.
multistrike : Om een contrastrijk plaatje te bekomen is het (zeker bij een ouder lint) interessant om dezelfde informatie meerdere keren over elkaar te drukken.
In de routine is een parameter voorzien die aangeeft hoe dikwijls de informatie moet geprint worden. MULTISTRIKE = 2 geeft meestal zeer behoorlijke plaatjes. Op te merken valt dat de tijd die nodig is om de screencopy te maken rechtsstreeks evenredig is met de MULTISTRIKE-parameter !

De routine is volledig in machinetaal geschreven, de snelheid van de afdruk wordt volledig bepaald door de gemiddelde snelheid van de printkop. Het programma is beschikbaar op audiocassette en kost 600 Bfr / 30 Gld.
Te bestellen op het redactieadres :
Mottaart 20, 3170 Herselt, betaling met eurocheque of overschrijving op :
AMRO-bank Baarle-Nassau 46.07.36.051.

Coördinatenzoeker



COORDINATENZOEKER

Dit programma tekent stippen op het scherm en geeft er de coordinaten van aan die ook kunnen geSAVEd worden.
Het is in de eerste plaats bedoeld voor het copieren van kaarten die onder vorm van transparant op het scherm worden gelegd.
Men kan zich evenwel ook andere toepassingen inbeelden.
Het is verder mogelijk verder te werken op een bestaand beeldscherm door dit programma te MERGEn met een grafisch programma.
Alsud kunnen controles uitgevoerd en verbeteringen of aanvullingen aangebracht worden.
Bij gebruik van dit programma verschijnt een raam dat de grenzen van het beschikbare scherm aangeeft. Het teken in de hoek herinnert aan de code om te saven: men tikt dan een 5 teken code in, waarna men de "SAVE" dient te bevestigen met "J".
De rechthoekige cursor beweegt snel over het scherm. Best kiest men eerst een vrije hoek van de tekening en tikt man op TAB.
Op deze plaats zullen telkens de coordinaten worden afgebeeld.
Om naar het secure werk over te gaan tikt men een lettertoets in: nu komt een kruis op het scherm waarvan het snijpunt blank is. Hier zal telkens de coordinaat worden bepaald. Dit punt wordt zwart als men triggert en de coordinaten verschijnen.
De praktijk heeft bewezen dat dit vizier niet te fijn mag zijn.
Is dit programma aan een ander toegevoegd, dan dient eventueel een gegeven gewijzigd, naargelang de sprites waarmee men origineel werkt; zie daarvoor het programma zelf.
Verschijnt het puntje niet precies in het vizier, dan wijst dit op deze laatste fout.
Wil men op een bepaald ogenblik vb van grenzen overgaan naar rivieren, dan kan door het intikken van een cijfer dit als een negatief coordinatenpaar tussenvoegen en zich naderhand zo beter orienteren.

Voor alle commentaar of vragen:
R.CEUTERICK GENT (091) 22 72 73

```
54321 IFI$=CHR$(27)THENCLEAR:G=1:GOSUB54373 ELSE 54378
54322 DATA  16,  16,  16, 238, 16,  16,  16,0
54323 DATA  0, 124,  68,  68,  68, 124,   0,  0
54324 RESTORE54322:Z=1:H=20:V=20:R=0:C=3*(G+2):CV=-1
54325 LINE(0,0)-(255,191),1,B
54326 CLOSE:OPEN"GRP:" AS #1
54327 PRESET(3,3):PRINT#1,"#▲▲▲▲▲▲"
54328 A$="":FORY=1TO8
54329 READN:A$=A$+CHR$(N):NEXT
54330 SPRITE$(Z)=A$:IFZ=1 THEN Z=9:GOTO54328
54331 STRIG(K)ON:SPRITEON
54332 SN=9:GOSUB54355
54333 B=STICK(K):ON B GOSUB54340,54341,54342,54343,54344,54345,54346,54347:GOSUB54351
54334 IFSTRIG(K) THENGOSUB54356
54335 I$="":I$=INKEY$:IFI$="#" THEN54365
54336 IFI$=CHR$(9)THENCH=H:CV=V:GOSUB54358:GOTO54333
54337 W=VAL(I$):IFW>0 THENH(R)=-W:V(R)=-W:PLAY"A":R=R+1:GOTO54333
54338 IFI$>CHR$(57)THENGOSUB54350
54339 GOTO54333
54340 V=V-SN/2:RETURN
54341 GOSUB54340
54342 H=H+SN/2:RETURN
54343 GOSUB54342
54344 V=V+SN/2:RETURN
54345 GOSUB54344
54346 H=H-SN/2:RETURN
54347 GOSUB54346:GOTO54340
54348 GOSUB54351:GOTO54333
54349 GOTO54349
54350 IFSN=9 THENSN=1 ELSE SN=9
54351 IFH<1 THENH=1
54352 IFH>255THENH=255
54353 IFV<1 THENV=1
54354 IFV>191 THENV=191
54355 PUTSPRITE1,(H-M,V-M-1),1,SN:RETURN
54356 H(R)=INT(H):V(R)=INT(V)
54357 LINE(H,V)-(H+1,V+1),1,BF
54358 IFCV<0 THEN54362
54359 LINE(CH-5,CV)-(CH+26,CV+30),11,BF
54360 CM=CV+8:GOSUB54363:PRINT#1,USING"###";H
```

```
54361 CM=CV+20:GOSUB54363:PRINT#1,USING"###";V
54362 R=R+1:RETURN
54363 CP$="BM"+STR$(INT(CH))+","+STR$(INT(CM)):DRAWCP$:RETURN
54364 LINE(1,1)-(80,12),11,BF:PRESET(3,3):RETURN
54365 GOSUB54364:PRINT#1,"▲?.....":T$=INPUT$(5)
54366 GOSUB54364:PRINT#1,T$;",▲J?":X$=INPUT$(1):IFINSTR("Jj",X$)=0 THENGOSUB54364:GOTO54324
54367 GOSUB54364:CLOSE:T$="CAS:"+T$:OPENT$ FOR OUTPUT AS #1:L=0
54368 X=L MOD 10:IF X=0 THEN PRINT#1,52000!+L;:PRINT#1,"DATA▲";
54369 IFL=R THENCLOSE:GOSUB54372:GOTO54324
54370 PRINT#1,H(L);",";V(L);:IFX = 9 THENPRINT#1,CHR$(13) ELSE PRINT#1,",";
54371 L=L+1:GOTO54368
54372 ERASEH,V
54373 DIM H(500),V(500):RESTORE54322:IFG=0 THEN M=2 ELSEM=6
54374 RETURN
54375 LINE(2,160)-(50,190),11,BF
54376 PRESET(3,166):PRINT#1,USING"###";H
54377 PRESET(3,180):PRINT#1,USING"###";V
54378 KEYOFF:COLOR 1,11,11:SCREEN1:WIDTH29
54379 CLS:PRINT"▲▲▲▲▲Coordinatenzoeker.▲▲▲▲▲▲——————————":PRINT"Voor▲instructies:▲tik▲<BS>":PRINT"
54380 PRINT"▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▲▲ONTHOUD:▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲Omschakeling▲vlugge/trage▲▲▲▲cursor:▲lettertoets."
54381 PRINT"▲▲Cijfertoets:▲neg.▲coord.":PRINT
54382 PRINT"▲▲Afb.▲coordinaten:▲<TAB>":PRINT
54383 PRINT"▲▲Save:▲▲▲#":PRINT
54384 PRINT"▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲":PRINT
54385 PRINT:PRINT"▲Kies▲voor▲werkwijze";SPC(39);"met▲pijltjes▲en▲<space>▲(1)▲▲of▲stick▲en▲actietoets▲▲(2)";SPC(31)
54386 K$=INPUT$(1)
54387 IFK$=CHR$(8)THEN54392
54388 K=VAL(K$):IFK<1 OR K>2 THEN54386
54389 K=K-1
54390 G=1:SCREEN2,G:GOSUB54373
54391 GOTO54324
54392 SCREEN0:WIDTH37:PRINT"Dit▲programma▲kan▲direct▲worden▲ge-▲▲bruikt▲door▲een▲transparant▲op▲het▲▲▲scherm▲te▲plaatsen.":PRINT
54393 PRINT"Tik▲<BS>▲voor▲instructies▲om▲vanuit▲▲een▲bestaand▲beeld▲te▲werken.":PRINT
54394 PRINT"Met▲stick▲of▲toetsen▲kan▲een▲cursor▲▲rap(rechthoek)▲of▲traag(kruisvizier)▲over▲het▲scherm▲worden▲bewogen.":PRINT
54395 PRINT"Deze▲coordinaten▲kunnen▲worden▲uitge-voerd.▲Tik▲#,▲waarna▲een▲titelstring▲kan▲worden▲gegeven":PRINT
54396 PRINT"Om▲de▲coordinaten▲op▲het▲scherm▲af▲tebeelden▲kies▲met▲de▲cursor▲de▲goede▲▲hoek▲en▲tik▲<TAB>.":PRINT
54397 PRINT"De▲coordinaten▲kunnen▲in▲reeksen▲op-▲gedeeld▲worden▲door▲een▲getal▲in▲te▲▲drukken:▲dan▲ontstaat▲telkens▲een▲▲▲▲negatief▲coordinatenpaar."
54398 I$=INKEY$:IFI$=""THEN54398
54399 IFI$=CHR$(8)THEN54401
54400 GOTO54378
54401 CLS:PRINT"Om▲op▲een▲bestaand▲beeld▲te▲▲werken.▲——————————————":PRINT
54402 PRINT"MERGE▲het▲programma▲met▲het▲▲hoofd-▲▲programma.":PRINT
54403 PRINT"Voeg▲in▲hoofdprogramma▲de▲volgende▲▲▲regel▲toe:":PRINT
54404 PRINT"I$=INKEY$:IFI$=CHR$(27)THEN▲54321":PRINT
54405 PRINT"Naargelang▲de▲sprites▲waarmee▲in▲het▲hoofdprograma▲wordt▲▲▲gewerkt,▲▲▲▲▲▲▲verander▲in▲regel▲54321▲deze▲waarde▲":PRINT
54406 PRINT"▲G=0▲voor▲gewone▲sprites,▲▲▲▲▲▲▲▲▲▲▲▲▲G=1▲voor▲vergrote▲sprites.":PRINT
54407 I$=INKEY$:IFI$=""THEN54407
54408 GOTO54392
```

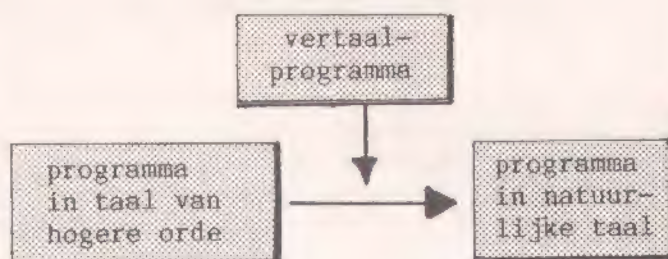


MACHINETAAL OP JE MSX-COMPUTER DE EERSTE STAPPEN

DEEL 1: Basisbegrippen en algemene beschrijving van de Z80A

1.1 Inleiding

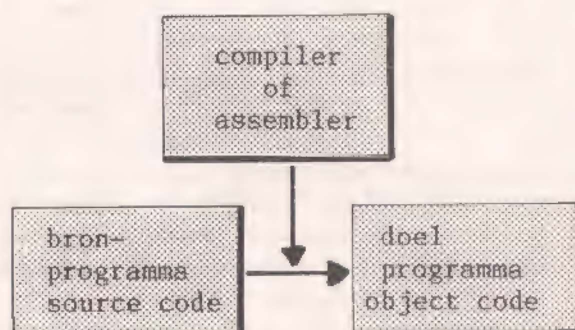
Je Msx-computer kan geprogrammeerd worden in meerdere talen. BASIC is hiervan de meest gebruikte. Dit komt omdat deze programmeertaal door de constructeur in je MSX-computer werd ingebouwd. Als je reeds enige ervaring hebt met BASIC-programmatie is het lezen en begrijpen van een BASIC-programma niet moeilijk meer. BASIC is immers een taal van hogere orde, waarmee bedoeld wordt dat ze ontwikkeld is om het programmeren voor de mens zo eenvoudig mogelijk te maken. Naast de talen van hogere orde bestaan er ook de talen van lagere orde. De natuurlijke taal van de computer,de binaire machinetaal,is daarvan het meest zuivere voorbeeld.Bovendien is deze natuurlijke taal de enige die de computer"echt begrijpt". Dit betekent dat een programma in welke andere taal het ook geschreven is, naar deze binaire machinetaal moet worden omgezet. Dit omzetten gebeurt door een vertaalprogramma.



Afb. 1 :Vertalen naar binaire machinetaal

Voor talen van hogere orde noemt men dit vertaalprogramma een compiler of een interpreter. De meeste BASIC-versies worden vertaald door middel van een interpreter.Talen van lagere orde worden vertaald door een assembler. Daar waar de opdrachten die gebruikt worden in een taal van hogere orde erg gelijken op een Engelse tekst, is dit voor een taal van lagere orde helemaal niet zo. In deze laatste zijn de op-drachten samengesteld met symbolische codewoorden, ook mnemomics genoemd. Deze mnemonics verwijzen nog slechts vaag naar de voorgestelde activiteit, zodat een programma geschreven in een symbo-lische machinetaal minder vlot leest.

Een symbolische machinegerichte taal wordt in de informatica een assembly-taal genoemd. Let op het verschil tussen een assembler en een assembly-taal. Indien de symbolen die kunnen gebruikt worden om de opdrachten van een programma samen te stellen uitsluitend de cijfers 0 en 1 zijn - de bouwstenen van het binair talstelsel - is het programma in de natuurlijke taal van de computer uitgeschreven. Deze programmanotatie wordt doelprogramma (object code) genoemd. De notatie in een hogere of lagere programmeertaal wordt bronprogramma (source code) genoemd.



Afb. 2 : Van source code naar object code

Programmeren in binaire machinetaal is gekkemanswerk. Elk door de mens geschreven machinetaalprogramma is daarom steeds ontwikkeld in een symbolische machinetaal. Leren programmeren in machinetaal eist dus in de eerste plaats een grondige kennis van de toegestane codewoorden. Het aantal, de opbouw en de betekenis van de codewoorden wordt bepaald door de microprocessor die in de computer is ingebouwd. In een MSX-computer is dit de Z80A van de firma Zilog. Omdat er een zeer nauw verband bestaat tussen de toegestane codewoorden en de ingebouwde processor onderstelt program-meren in machinetaal ook een voldoende inzicht in de opbouw en de werking van deze processor. Door middel van deze artikelenreeks zal gepoogd worden om deze twee fundamentele stappen bij het leren programmeren in machinetaal te verduidelijken.

1.2 Het MSX-microcomputersysteem

Schematisch kan de opbouw van een MSX-computer als volgt worden voorgesteld.



Afb. 3 : Basiscomponenten van een computersysteem

Het belangrijkste invoerorgaan van je MSX-computer is het toetsenbord en het meest gebruikte uitvoerorgaan het beeldscherm. In het geheugen van een computer worden programma's, gegevens en resultaten ge-stockeerd. Na compileren of assembleren zijn al deze elementen omgezet in een rij van nullen en enen. Algemeen noemt men zulke rij een bitrij. Een computergeheugen moet dus essentieel opgebouwd zijn uit componenten waarmee slechts twee verschillende waarden kunnen worden voorgesteld. We laten hier buiten beschouwing met welke electronische middelen dit gebeurt en stellen deze twee verschillende waarden algemeen voor door 0 en 1. Deze twee cijfers vormen de basis van het binair talstelsel. De Engelse benaming voor een binair cijfer is binary digit. Omwille van de belangrijkheid van de binaire getallen ,vooral in de beginperiode van de informatica, werd de benaming binary digit in 1947 door Shannon en Tuckey samengetrokken tot het letterwoord BIT. Een bit is bijgevolg steeds een van de twee binaire cijfers 0 of 1. Je kan dus ook zeggen dat een computer-geheugen opgebouwd is uit een groot aan-tal componenten die allemaal een bit kunnen bevatten. Om practische redenen worden deze componenten meestel per 8 gegroepeerd. Ze vormen samen een geheugencel. In een geheugencel kunnen dus 8 bits worden opgeborgen. Een groep van 8 bits noemt men een byte. De Nederland-se benaming voor een byte is octade. Het Engelse woord byte is echter zo ingeburgerd dat we deze benaming blijven gebruiken. Het aantal geheugencellen dat tot een computergeheugen behoort,bepaalt de capaciteit van dat geheugen.

De geheugencapaciteit wordt uitgedrukt in het aantal bytes die in dat geheu-gen kunnen gestockeerd worden. Een veel voorkomende capaciteit bij microcomputers is 65536 bytes. Omdat gans de computerorganisatie gebaseerd is op het binair talstelsel,gebruikt men geen veelvoud van 10 maar van 2 als eenheid om grotere capaciteiten te noteren. Zo is 1 Kb (kilobyte) gelijk aan 1024 bytes.

Noteer dat $1024 = 2^{10}$

Een computergeheugen waarin 65536 bytes kunnen gestockeerd worden, heeft bijge-volg een capaciteit van 65536:1024=64Kb. Om een onderscheid te kunnen maken tus-sen de verschillende geheugencellen wor-den ze genummerd. Voor een 64Kb-geheugen gaat deze nummering van 00000 tot 65535. Het nummer dat bij een bepaalde geheugen-cel hoort noemt men zijn adres. De belangrijkste component van een computersysteem is de centrale verwerkingseenheid (CVE). Hierin geschieden alle verwerkingen die tijdens een programma-uitvoering nodig zijn: rekenkundige bewerkingen, vergelijkingen, transport van opdrachten, gegevens en resultaten... Het hart van de CVE van een MSX-computer is de Z80A-microprocessor.

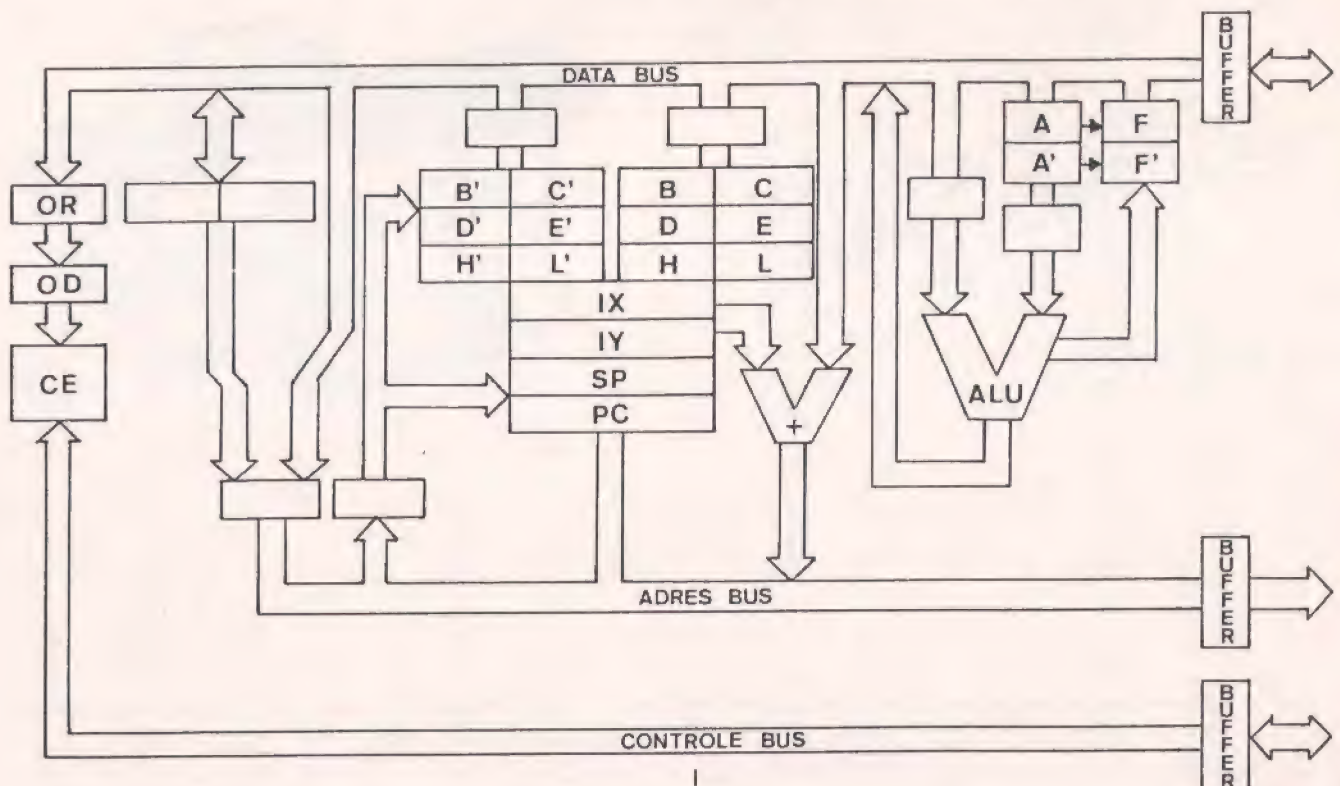
1.3 De Z80A-microprocessor

De eigenlijke programma-uitvoering in een microcomputer geschiedt in de ingebouwde microprocessor. Een microprocessor is een complexe electronische schakeling die een groot aantal verschillende taken kan verrichten. Essentieel kan een microprocessor opgesplitst worden in drie delen : een controleblok, een rekeneenheid en een transportsysteem. De belangrijkste taak van het controle-blok is ervoor zorgen dat de opdrachten van een programma in de correcte volg-orde worden uitgevoerd.Het controblok van de Z80A-processor bestaat daartoe uit een aantal componenten waarvan de programcounter (PC), het opdrachtregis-ter (OR),de opdrachtdecoder(OD) en de controle-eenheid (CE) de belangrijkste zijn. De programcounter (PC) bevat op elk ogenblik het adres van de geheugencel waarin de volgende opdracht (of het eerste deel ervan) die moet worden uitgevoerd, zit opgeborgen.

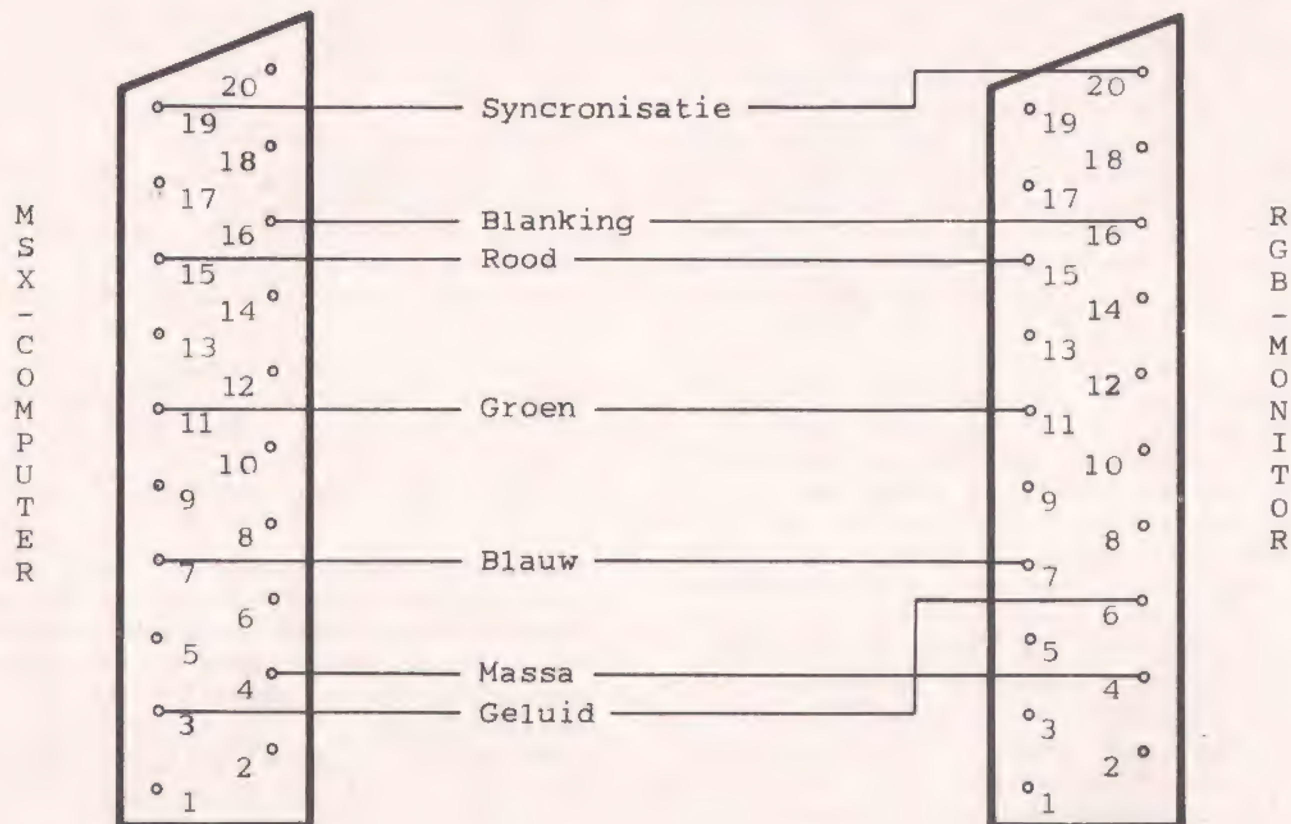
Door middel van de programcounter weet de microprocessor dus op elk ogenblik in welk stadium de programmaverwerking zich bevindt. Voordat een opdracht kan uitgevoerd worden moet ze vanuit het geheugen naar de microprocessor worden overgebracht. Ze wordt hiertoe gekopieerd in het opdrachtregister (OR) van de microprocessor. Daarna wordt ze in de opdrachtdecoder (OD) geanalyseerd. Deze decodeer-schakeling bepaalt welke activiteit de opdracht voorstelt, met welke gegevens ze moet worden uitgevoerd en wat er met een eventueel resultaat moet gebeuren. Na deze analyse zitten in de controle-eenheid de twee componenten waarin de opdracht werd opgesplitst. We noemen ze de operatiecode (wat moet er gebeuren) en de operand (waarmee of waarop moet de operatiecode werken). De nodige signalen om de opdracht uit te voeren worden daarna door de controle-eenheid (CE) gegenereerd. Naast het opvolgen van de programma-uitvoering staat de microprocessor ook in voor het uitvoeren van de bewerkingen. De voornaamste component hiervoor is de rekeneenheid , voorgesteld door ALU. De rekeneenheid heeft ook een aantal hulpgeheugens ter beschikking om zijn taak vlot te kunnen uitvoeren. Een hulpgeheugen in een microprocessor noemt men algemeen een register. Het wordt vooral gebruikt om tussentijdse resultaten tijdelijk te bewaren. De belangrijkste registers van de rekeneenheid zijn :

- de enkelvoudige registers :A,B,C,D, E,H,L,F - de registerparen : BC,DE,HL,IX,IY,SP De enkelvoudigeregisters zijn 1 byte lang;deregisterparen 2. Het transport tussen de verschillende componenten van de microprocessor wordt georganiseerd door middel van de interne busstructuur. Deze wordt in functioneel opzicht opgesplitst in drie delen:
  - de DATA-bus
  - de ADRES-bus
  - de CONTROLE-bus Via de DATA-bus worden de opdrachten,de gegevens en de resultaten verplaatst.Om-dat de DATA-bus bestaat uit 8 parallelle lijnen die elk 1 bit kunnen transporte-ren,geschiedt een verplaatsing via de DATA-bus steeds per byte (=8 bits). Een microprocessor waarvan de DATA-bus uit 8 bitlijnen bestaat,wordt een 8-bitproces-sor genoemd.

Elk adrestransport tijdens een programmaverwerking gebeurt langs de ADRES-bus. Meestal bevat de ADRES-bus 16 bitlijnen. We zullen in een volgend artikel verklaren waarom dit zo is. Alle controlesignalen tenslotte worden via de CONTROLE-bus doorgestuurd. Het nut en de specifieke taak van de hulpregisters zal doorheen de artike-lenreeks verduidelijkt worden voor zo-ver dit nodig is in functie van het le-ren programmeren in de symbolische machinetaal van de Z80A-microprocessor. Als alle componenten van de Z80A in een schema worden samengebracht,ontstaat volgende afbeelding.



MSX-RGB aansluiting



karakterset BROTHER DAISYWHEEL ASCII-schijf

32	#20		33	#21	!	34	#22	"	35	#23	#
36	#24	$	37	#25	%	38	#26	&	39	#27	'
40	#28	(	41	#29	)	42	#2A	*	43	#2B	+
44	#2C	,	45	#2D	-	46	#2E	.	47	#2F	/
48	#30	0	49	#31	1	50	#32	2	51	#33	3
52	#34	4	53	#35	5	54	#36	6	55	#37	7
56	#38	8	57	#39	9	58	#3A	:	59	#3B	;
60	#3C	<	61	#3D	=	62	#3E	>	63	#3F	?
64	#40	@	65	#41	A	66	#42	B	67	#43	C
68	#44	D	69	#45	E	70	#46	F	71	#47	G
72	#48	H	73	#49	I	74	#4A	J	75	#4B	K
76	#4C	L	77	#4D	M	78	#4E	N	79	#4F	O
80	#50	P	81	#51	Q	82	#52	R	83	#53	S
84	#54	T	85	#55	U	86	#56	V	87	#57	W
88	#58	X	89	#59	Y	90	#5A	Z	91	#5B	[
92	#5C	\	93	#5D	]	94	#5E	^	95	#5F	_
96	#60	`	97	#61	a	98	#62	b	99	#63	c
100	#64	d	101	#65	e	102	#66	f	103	#67	g
104	#68	h	105	#69	i	106	#6A	j	107	#6B	k
108	#6C	l	109	#6D	m	110	#6E	n	111	#6F	o
112	#70	p	113	#71	q	114	#72	r	115	#73	s
116	#74	t	117	#75	u	118	#76	v	119	#77	w
120	#78	x	121	#79	y	122	#7A	z	123	#7B	{
124	#7C	|	125	#7D	}	126	#7E	~			

"Kan ik jouw wiskunde een keer overnemen?"

"Als jullie thuis een Philips MSX hebben wèl!!!"...





Daar sta je dan. Wil je je achterstand in je huiswerk wat inlopen na je dagje schoolziek krijg je zo'n antwoord!

Je had het nog zo tegen je ouders gezegd toen er sprake was van een homecomputer maar ze moesten het weer even beter weten natuurlijk.

"Die MSX van Philips," zei je, "heeft een verdomd goeie microsoftware die BASIC versterkt is."

Je ouders maar kijken.

"En er kan van alles en nog wat op aangesloten worden," zei je verder, "een heleboel videospelletjes," en toen sprong je buurjongen die bij je thuis was, je bij, want die hebben er al een en daar zit je tegenwoordig dan ook vaak.

"Ja," zei die, "mijn vader maakt zijn eigen programma's, want in die MSX van Philips zit een geheugen van 48 K of 80 K RAM en 32 K ROM."

Je moest wel even lachen om zijn geleerde gezicht, maar je vader zei verbaasd "Oh, en hoe oud is je vader dan wel?"

Stomme vraag toch! Je vader is net zo oud als die van je vriend. Maar daar komt hij altijd mee aanzetten. Dat hij te oud is om al die dingen te leren.

Zo heb je een tijd staan argumenteren. Over het spotgemakkelijke toetsenbord met 73 stevige toetsen die tegen een stootje kunnen, zodat er niet altijd tegen je jongste broertje "Afblijven!" geroepen moet worden.

Je had het over de 16 kleuren die mogelijk zijn, de tekenset met 254 alfanumerieke en grafische tekens, de 256 figuurtjes van de videoprocessor.

Over de geluidsgenerator met de drie toonreeksen van telkens 8 octaven, zodat je bij je zelfgeschreven programma's desnoods nog muziek kunt maken ook.

Philips MSX homecomputers zijn tegenwoordig eigenlijk doodgewoon, alleen je ouders moeten er nog aan wennen dat leren nu veel spelenderwijzer gaat dan in hun eigen jeugd.

Je vader was alleszins zichtbaar onder de indruk en toen je moeder vroeg wat een Philips MSX kost, wist je dat het goed zat.

"Goedkoop," zei je daarom extra overtuigend en daar had je gelijk in. Nu nog trouwens, vooral als het bij je thuis duidelijk is dat je er zowat alle software en hardware van zo'n 30 andere merken op kunt aansluiten. En dan is jullie gezin natuurlijk vertrokken voor de toekomst!!

Loop eens binnen bij ons. Ons adres staat hieronder. We hebben een folder voor je en daarmee kun je je ouders het nog eens uitleggen. Maar je hebt best kans dat dat niet meer nodig is en dat ik ze net een Philips MSX heb verkocht.



PHILIPS MSX HOME COMPUTER SYSTEM



PHILIPS

DE TECHNOLOGIE STAAT NIET STIL... PHILIPS LOOPT VOOROP

Sony Plotter-printer

deel 2

6. Wijzigen van het karakterformaat tijdens de programmaverwerking

a. Met een expliciete code

In nr 4 hebben we aangegeven dat de opdracht LPRINT CHR$(&H12)+"N" het karakterformaat van de programmalis-ting instelt in formaat N. Hierbij is N een geheel getal gelegen tussen 0 en 15. De hexadecimale notatie &H12 kan ook decimaal worden opgegeven. De opdracht wordt dan : LPRINT CHR$(18). Door deze opdracht in te bouwen in een programma worden de programmaresultaten in het geselecteerde formaat afgedrukt. Als voorbeeld het programma in afbeel-ding 8. Laat het uitvoeren met je eigen adres als invoergegeven en je bekomt een analogon van afbeelding 9 als resultaat.

```
10 CLS
20 INPUT "naam        : ";N$
30 INPUT "straat      : ";S$
40 INPUT "woonplaats: ";W$
50 LPRINT CHR$(18)+"2"
60 LPRINT N$
70 LPRINT S$
80 LPRINT W$
90 LPRINT:LPRINT
100 LPRINT CHR$(18)+"5"
110 LPRINT N$
120 LPRINT S$
130 LPRINT W$
```

Afb. 8 : Karakterformaat wijzigen vanuit het programma

```
MSX-club
Mottaart 20
3170  Herselt
MSX-club
Mottaart 20
3170  Herselt
```

Afb. 9 : Resultaat van het programma uit afbeelding 8

In programmaregel 50 van afbeelding 8 wordt het karakterformaat op 2 gezet. Het adres wordt vervolgens een eerste maal in dit formaat afgedrukt. Programmaregel 100 wijzigt dit formaat in 5.

b. Vanuit een stringvariabele

Het LPRINT-sleutelwoord waarmee het karakterformaat wordt vastgelegd, moet gevolgd worden door een string. Deze string bestaat uit twee delen : de stuurcode CHR$(18) en de formaatcode "5". Beide strings worden samengevoegd (geconcateneerd) door middel van het +teken (de concatenator) tot eén string: CHR$(18)+"5". Het is echter ook mogelijk deze door een kommapunt te scheiden : CHR$(18);"5". Door deze formaatcode toe te kennen aan een stringvariabele kan het karakterfor-maat op een interactieve wijze worden opgegeven. Het programma in afbeelding 10 laat toe een adres in te tikken en vraagt daarna in een FOR-NEXT-lus drie-maal naar een karakterformaat (regel 60)

```
10  CLS
20  INPUT "naam        : ";N$
30  INPUT "straat      : ";S$
40  INPUT "woonplaats: ";W$
50 FOR I=1 TO 3
60 INPUT "karakterformaat 0-15 ";F$
70 LPRINT CHR$(18)+F$
80 LPRINT N$
90 LPRINT S$
100 LPRINT W$
110 LPRINT :LPRINT :LPRINT
120 NEXT I
```

Afb. 10 : Het karakterformaat vanuit een stringvariabele

Afbeelding 11 geeft het ingevoerde adres afgedrukt in respectievelijk formaat 0,1 en 2. Merk op dat ook het interligne (resultaat van een lege LPRINT-opdracht; programmaregel 110) beinvloed wordt door het karakterformaat.

7. Wijzigen van de kleurcode tijdens de programmaverwerking

De algemene formulering van de opdracht om een kleur te selecteren is :

regelnummer LPRINT CHR$(27)+"CN" De string in deze LPRINT-opdracht be-staat ook hier uit twee delen : de stuurcode CHR$(27) en de string die de geselecteerde kleur aanwijst : "CN" De hoofdletter C is hier noodzakelijk en de hoofdletter N moet vervangen wor-den door een cijfer van 0 tot en met 3. Door gebruikt te maken van de concatena-tor kan ook hier weer de kleurcode tij-dens de programmaverwerking gewijzigd worden. Afbeelding 16 geeft een program-ma waarmee deze mogelijkheid geillus-treerd wordt. De lezer die over een SONY PRN-C41 plotter-printer beschikt zal merken dat de resultaten in drie ver-schillende kleuren op de printer worden afgedrukt. Het programma werd echter zo aangevuld dat de resultaten ook op het beeldscherm worden afgedrukt. Op deze wijze zijn de programma's ook geschikt voor een lezer die geen SONY-plotter-printer heeft. In dit geval kunnen zelfs alle opdrachten waarvan het sleutelwoord LPRINT is worden weggelaten. We zullem dit principe daar waar het zinvol is ook in de volgende artikels blijven vol-gen. Dit leidt uiteraard tot langere programma's. We vermoeden echter er een ruimer MSX-publiek mee te plezieren. Merk tussendoor ook de nauwkeurigheid op waarmee je MSX-computer rekent.

We geven van dit programma geen verwer-king omdat de doelstelling ervan, nl. een 3-kleuren afdruk maken, in dit mo-nochromatische tijdschrift niet kan be-reikt worden. Het programma in afbeelding 16 selec-teert de kleur door de expliciete kleur-code in de LPRINT-opdracht op te nemen. Het is duidelijk dat ook deze code , net zoals de karakterformaatcode, door mid-del van een stringvariabele met de stuurcode (CHR$(27)) kan geconcateneerd worden. Afbeelding 17 als illustratie.

```
10 CLS
20 INPUT "tik een zin in ";Z$
30 PRINT "zwart  : 0"
40 PRINT "blauw  : 1"
50 PRINT "groen  : 2"
60 PRINT "rood   : 3"
70 INPUT "afdrukkleur (0-3) : ";K$
80 LPRINT CHR$(27)+"C"+K$
90 LPRINT Z$
```

Afb. 17 : De kleurcode via een string-variabele

Afb. 16 : De kleurcode aanpassen tij-dens de programmaverwerking

```
10 CLS
20 PRINT "getal";TAB(7);"kwadraat";TAB(18);"vierkantswortel"
30 PRINT
40 LPRINT CHR$(27)+"C1"
50 LPRINT CHR$(11);
60 LPRINT "getal",
70 LPRINT CHR$(27)+"C2"
80 LPRINT CHR$(11);
90 LPRINT "kwadraat",
100 LPRINT CHR$(27)+"C3"
110 LPRINT CHR$(11);
120 LPRINT "vierkantswortel"
130 LPRINT
140 FOR I=0 TO 9
150 PRINT TAB(1);I;TAB(9);I*I;TAB(17);SQR(I)
160 LPRINT CHR$(27)+"C1"
170 LPRINT CHR$(11);
180 LPRINT I,
190 LPRINT CHR$(27)+"C2"
200 LPRINT CHR$(11);
210 LPRINT I*I,
220 LPRINT CHR$(27)+"C3"
230 LPRINT CHR$(11);
240 LPRINT SQR(I)
250 NEXT I
```

ABCDEFGHIJKLMNOP

Piet Luttig
Ministraat 00
12345 NIETIG

Piet Luttig
Ministraat 00
12345 NIETIG

Piet Luttig
Ministraat 00
12345 NIETIG

Afb. 11 : Resultaat van het programma uit afbeelding 10

In programmaregel 70 wordt de stringvariabele F$ geconcateneerd met de stuurcode CHR$(18). Deze nieuwe string bepaalt het correcte karakterformaat. Met wat kennis van BASIC-programmatie zijn we nu in staat om de teller van een FOR-NEXT-lus als karakterformaat te gebruiken. We moeten er dan wel voor zorgen dat deze teller , die van het numerieke type is, omgezet wordt in een stringvariabele. Dit kan gebeuren met de BASIC-functie : STR$(). Afbeelding 12 bevat het programma dat afbeelding 13 als resultaat geeft.

```
10 CLS
20 LPRINT CHR$(18)+"15"
30 FOR I=15 TO 0 STEP-1
40 F$=STR$(I-1)
50 B$=CHR$(80-I)
60 LPRINT B$;
70 LPRINT CHR$(18)+F$
80 LPRINT CHR$(11);
90 NEXT I
```

Afb. 12 : Het karakterformaat vanuit een FOR-NEXT-teller

Afb. 13 : Resultaat van het programma uit afbeelding 12

Programmaregel 40 zet de waarde van de numerieke variabele I om in een stringnotatie F$. Deze kan dan gebruikt worden in de string die het karakterformaat vastlegt. Nieuw in dit programma is ook de LPRINT-opdracht met de stuurcode CHR$(11). Deze code stuurt de tekenpen een regel naar boven.Deze opdracht is nodig omdat de opdracht :

LPRINT CHR$(18)+F$ de pen een regel naar beneden plaatst. Regel 70 afsluiten met eem kommapunt faalt. Afbeeldingen 14 en 15 illustreren wat mogelijk is indien het karakterformaat via de RND-functie wordt bepaald. Programmaregel 20 zorgt ervoor dat de randomgenerator bij elke nieuwe programma-uitvoering op een andere plaats start, zodat steeds een andere rij willekeurige getallen wordt gegenereerd.

```
10 CLS
20 A=RND(-TIME)
30 INPUT "tik een zin in ";Z$
40 LPRINT CHR$(18)+"10"
50 FOR I=1 TO LEN(Z$)
60 G=INT(10*RND(1))
70 F$=STR$(G)
80 L$=MID$(Z$,I,1)
90 LPRINT L$;
100 LPRINT CHR$(18)+F$
110 LPRINT CHR$(11);
120 NEXT I
```

Afb. 14 : Random karakterformaat

Ik leer programmeren

Afb. 15 : Resultaat van het programma uit afbeelding 14

Indien in programmaregel 70 een getal wordt opgegeven dat groter is dan 3, wordt automatisch de bewerking modulo 4 uitgevoerd. Dit betekent dat de rest bepaald wordt van de deling van dat getal door 4. Deze rest is steeds 0,1, 2 of 3.Je kan ook stellen dat de toekenning van de kleurcode cyclisch doorloopt over de rij van de natuurlijke getallen:

0	1	2	3	4	5	6
zwart	blauw	groen	rood	zwart	blauw	groen

Merk op dat de string in programmaregel 80 uit drie delen bestaat : de stuurcode CHR$(27), de string "C" die verwijst naar de activiteit "kleur kiezen" en de kleurcode , opgegeven via K$. Indien de concatenator vervangen wordt door een kommapunt moet programmaregel 80 als volgt geformuleerd worden :

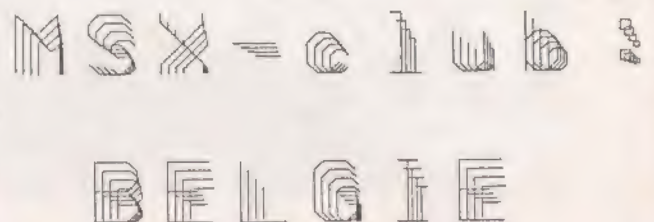
```
80 LPRINT CHR$(27);"C";K$
```

8.Enkele speciale stuurcodes voor de tekstmode

Als slot van de bespreking van de mogelijkheden van de SONY PRN-C41 als printer vermelden we nog enkele stuurcodes die soms handig zijn om bepaalde effecten te bekomen. De code CHR$(8) in een LPRINT-opdracht plaatst de tekenpen 1 letterteken terug. Hierbij moet worden opgemerkt dat dit terugplaatsen functie is van het op dat ogenblik ingestelde karakterformaat. De code CHR$(10) doet het papier 1 regel opschuiven en de code CHR$(12) schuift het papier op tot 297 mm onder het be-ginpunt van het afdrukken. Afbeeldingen 18 en 19 als illustratie van de CHR$(8) als struurcode.

```
10 CLS
20 LPRINT CHR$(18)+"10"
30 INPUT "woord";W$
40 L=LEN(W$)
50 FOR J=1 TO L
60 LPRINT TAB(3);
70 FOR I=6TO 3 STEP -1
80 F$=STR$(I)
90 LPRINT CHR$(18);F$
100 LPRINT CHR$(11);
110  LPRINT CHR$(8);
120 LPRINT MID$(W$,J,1);
130 NEXT I
140 NEXT J
```

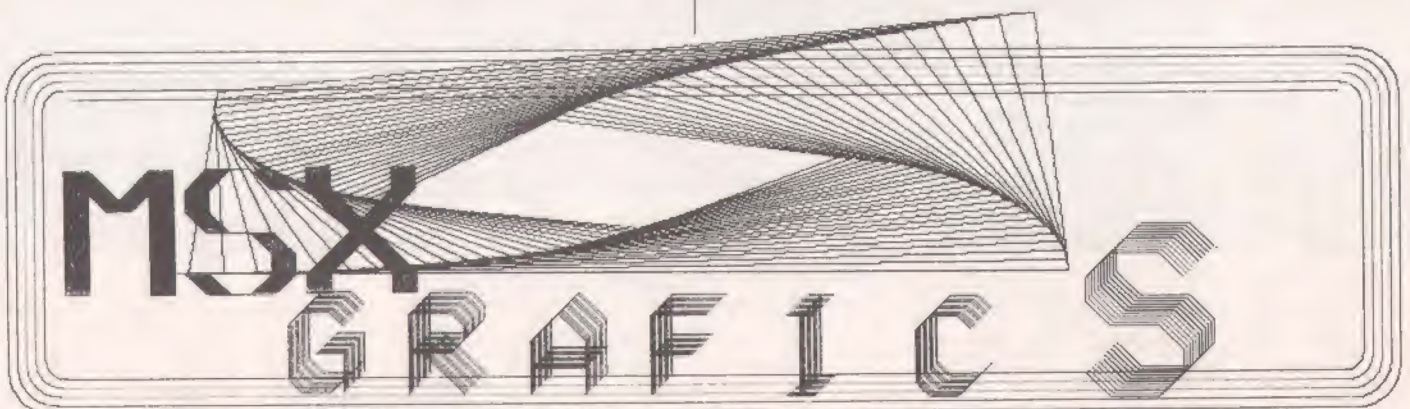
Afb. 18 : CHR$(8) als code



Afb. 19 : Resultaat van het programma uit afbeelding 18

(wordt vervolgd)

Bruno Van Rompaey



Een toemaatje van onze grafische medewerker Jos Meulenbergs



```
1Ø ' ****************
2Ø ' ** LOGO  M S X **
3Ø ' ** B Y   W I M **
4Ø ' ** DEWIJNGAERT **
5Ø ' ****************
6Ø COLOR15,1,1:CLS:REM ZET KLEUR
7Ø SCREEN2:REM GRAFISCH SCHERM
```

```
8Ø REM TEKEN ROUTINE
```

```
9Ø LINE (1Ø,13Ø)-(3Ø,7Ø)
1ØØ LINE (3Ø,7Ø)-(5Ø,7Ø)
11Ø LINE (5Ø,7Ø)-(55,9Ø)
12Ø LINE (55,9Ø)-(6Ø,7Ø)
13Ø LINE (6Ø,7Ø)-(8Ø,7Ø)
14Ø LINE (8Ø,7Ø)-(9Ø,115)
15Ø LINE (9Ø,115)-(125,115)
16Ø LINE (125,115)-(115,11Ø)
17Ø CIRCLE(115,9Ø),2Ø,15,.7854,4.7124,1.
    1
18Ø LINE (11Ø,7Ø)-(17Ø,7Ø)
19Ø LINE (17Ø,7Ø)-(185,9Ø)
2ØØ LINE (185,9Ø)-(2ØØ,7Ø)
21Ø LINE (2ØØ,7Ø)-(22Ø,7Ø)
22Ø LINE (22Ø,7Ø)-(2ØØ,1Ø5)
23Ø LINE (2ØØ,1Ø5)-(22Ø,13Ø)
24Ø LINE (22Ø,13Ø)-(195,13Ø)
25Ø LINE (195,13Ø)-(185,115)
26Ø LINE (185,115)-(17Ø,13Ø)
27Ø LINE (17Ø,13Ø)-(145,13Ø)
28Ø LINE (145,13Ø)-(17Ø,1Ø5)
29Ø LINE (17Ø,1Ø5)-(16Ø,9Ø)
3ØØ LINE (16Ø,9Ø)-(12Ø,9Ø)
31Ø LINE (12Ø,9Ø)-(12Ø,95)
32Ø CIRCLE(125,112.75),2Ø,15,3.927,2.356
    2,.9
33Ø LINE (12Ø,13Ø)-(8Ø,13Ø)
34Ø LINE (8Ø,13Ø)-(7Ø,11Ø)
35Ø LINE (7Ø,11Ø)-(6Ø,13Ø)
36Ø LINE (6Ø,13Ø)-(45,13Ø)
37Ø LINE (45,13Ø)-(4Ø,11Ø)
38Ø LINE (4Ø,11Ø)-(3Ø,13Ø)
39Ø LINE (3Ø,13Ø)-(1Ø,13Ø)
```

```
4ØØ REM VUL LETTERS OP
```

```
41Ø PAINT(6Ø,1ØØ),15,15
42Ø PAINT(15Ø,8Ø),15,15
43Ø PAINT(14Ø,12Ø),15,15
```

```
44Ø REM TEKEN BOORD
```

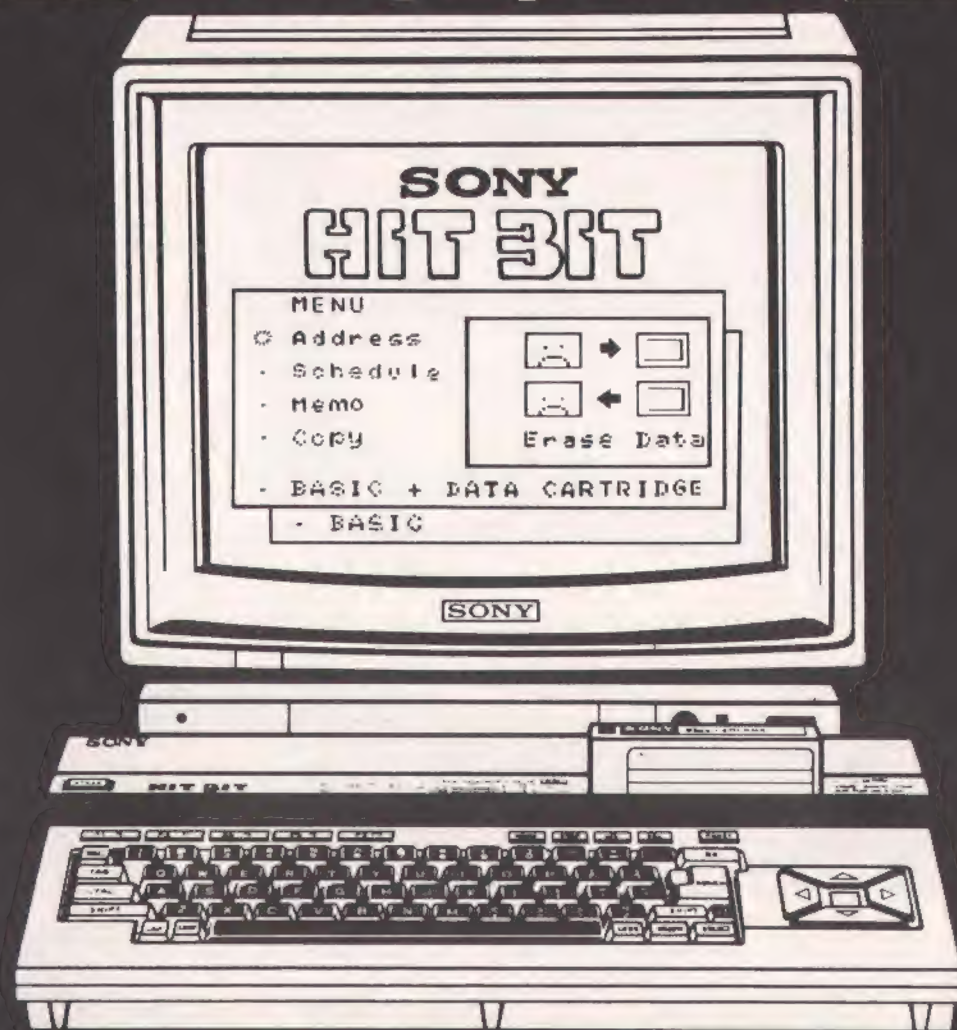
```
45Ø FORT=1TO1ØØØ
46Ø   FORC=1TO15
47Ø   LINE (Ø,Ø)-(256,6Ø),C,BF
48Ø   LINE (Ø,14Ø)-(256,192),C,BF
49Ø    FORW=1TO1ØØØ
5ØØ    NEXTW
51Ø   NEXTC

52Ø NEXTT

53Ø GOTO53Ø
```



Hit Bit MSX, vergelijken is hem kiezen



Sony Hit Bit, met MSX standaard vervult al uw behoeften.
Beschikbaar met 32 K RAM aktief geheugen (model HB 55) of met 80 K RAM (model HB75), 16 K video-RAM incl.

Door zijn kwaliteit en de talrijke mogelijkheden die hij biedt is de Sony Hit Bit de familiale komputer bij uitstek. Als exclusiviteit voor Sony : de ingebouwde persoonlijke databank vergemakkelijkt het beheer van gegevens die u dagelijks nodig heeft : afspraken, adressen telefoonnummers en de talrijke nuttige gegevens.

Vraag ons nu aanvullende inlichtingen.

SONY®

Manic Miner

```
*********************************************
*  TITEL   : Manic Miner                    *
*  AARD    : avonturentocht                 *
*  LADEN   : load "MANIC",r    TIJD : 4'30" *
*  STARTEN : spatiebalk                     *
*********************************************
```

1.Besturing

- bewegen links : S,F,H,K of cursor links
  bewegen rechts : D,G,J,L of cursor rechts
- springen : shift of cursor omhoog
- geluid harder : P
  geluid stiller : Q
- beeld stil : W
  beeld bewegen : bewegingstoets
- afbreken spel : STOP-toets

2.Doel van het spel

Jij speelt de rol van Willy de mijnwerker.In een vroegere beschaving werd dit zware werk door robots opgeknapt.
Deze beschaving stierf echter uit, maar de robots bleven verder werken en vergaarden een enorme hoeveelheid kostbaar materiaal.
Slimme Willy kwam toen op het illustere idee om deze schat, die volgens zijn idee zo maar voor het rapen lag, te gaan opzoeken.
Daarvoor moet hij echter door 20 grotten waar heel wat moeilijkheden op hem wachten.
In elke grot moet hij de blinkende voorwerpen (sleutels, sneeuwschoenen, bananen....) gaan halen vooraleer hij verder kan.
Er moet vooral opgepast worden voor de robots maar ook voor giftige spinnen, puntige voorwerpen, drijfzand en andere obstakels.
Elk blinkend voorwerp brengt hem 100 punten op.
Bij het bereiken van de volgende grot (via de mijningang) krijgt men bonuspunten afhankelijk van de hoeveelheid lucht die opgebruikt werd.(witte band onderaan het scherm)
Bij elke 10.000 punten krijgt men een extra Willy .
In de 20 ste grot, "The final barrier " genoemd kan men eindelijk de schat vinden.

3.Kritische beschouwingen
----

a)schermopbouw
- Willy lijkt niet erg op een mijnwerker en de robot in het eerste scherm is meer een mislukte kikvorsman ; nochtans stoort dit geenszins het spel.
- De gevaren waaraan je blootgesteld wordt zijn soms zeer verrassend en mooi getekend hoewel men er soms niet direkt een naam aan kan geven.
- Men moet dus 20 grotten (20 schermen) elk met zijn eigen naam en karakteristieken doorlopen.
- Wanneer men na het laden het spel niet onmiddellijk start krijgt men na een tijdje een overzicht van de 20 grotten.Echt knap !

b)geluid
- Op de achtergrond klinkt steeds een vrij monotoom wijsje.

c)betrokkenheid
- Het spel is vrij amusant, en hoe langer je speelt hoe groter de drang wordt om steeds dieper door te dringen , op jacht naar de schat.
- Het grote probleem is wel dat men slechts over 3 Willy's beschikt en wanneer die gepakt zijn moet je terug starten bij grot 1.
- Wanneer men 10.000 punten vergaard heeft krijgt men wel een extra Willy.(hiervoor moet men echter wel 6 grotten doorlopen)

d)moeilijkheidsgraad
- In de beginfase lijkt het nogal moeilijk om de volgende grot te kunnen bereiken, maar na een tijdje oefenen vlot dit goed.
- Nochtans moet men in elke grot de juiste weg weten uit te stippelen om erdoor te komen. Dit vraagt voor elke nieuwe grot toch een groot aantal Willy's.
- Blijf je echtergeconcentreerd en vastberaden dan lukt het je zeker om tot de schat door te dringen.

e)besturing
- De grote keuze uit de bedieningstoetsen maakt het mogelijk om de voor jou meest geschikte toetsen uit te kiezen.

4.Applausmeter

```
-----------------------------------------------
!zeer  !zwak !rede !matig!goed !zeer !super !
!zwak  !     !lijk !     !     !goed !      !
!      !     !     !     !     !     !      !
!#####!#####!#####!#####!##    !     !      !
-----------------------------------------------
```



SPECTRAVIDEO

SVI 728
SVI 738

PERSONAL...



& PROFESSIONAL

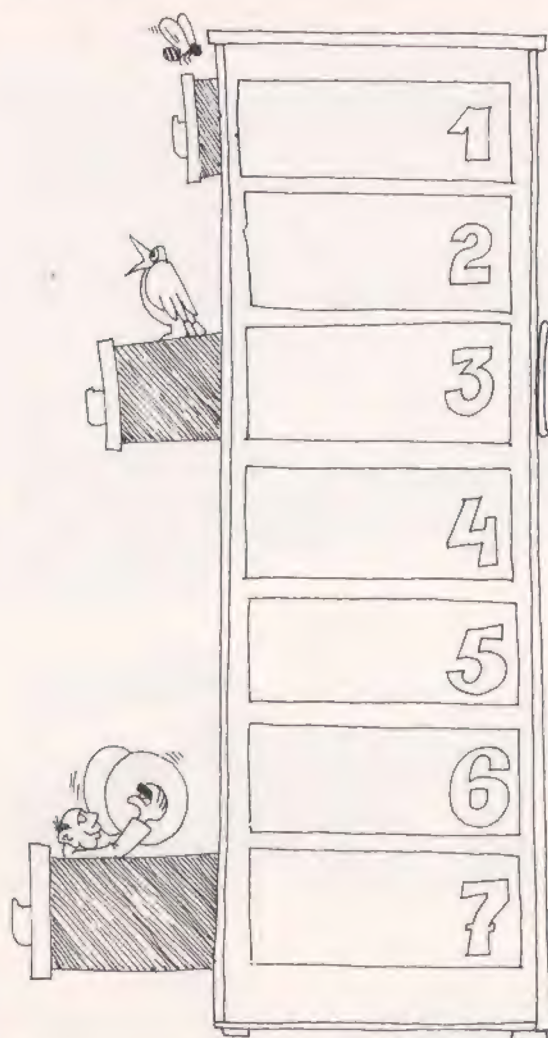


Hoofdverdeler :

serka pvba

computers-kantoorautomatisering
Dorpsstraat 89, 3150 BOOISCHOT
015/22.37.10 - 22.24.84
Dealerlijst op aanvraag

Soundmaster





Met dit programma kan je op een eenvoudige manier de klankrijkdom van MSX gaan verkennen.

Nadat het programma correct is ingetikt (of probleemloos geladen van de verzamelcassette),kan je met de CURSOR-toetsen links/rechts een register selecteren. Met de CURSOR-toetsen OP/NEER kan je dan de inhoud van dit register wijzigen.

Als een bepaald klankeffect je bevalt, dan kan je de waarden voor de registers noteren, en later in uw eigen programma's gebruiken.

```
10 REM SOUND-MASTER
20 REM ------------
30 REM LX()= X-POSITIE
40 REM LY()= Y-POSITIE
50 REM S$()= SOUND - NAAM
60 REM MV()= MAXIMUM - WAARDE
70 REM IV()= IN & ACT WAARDE
```

```
80 P$=CHR$(207):REM ▶
90 DEFINT A-Z
95 KEY OFF
```

```
100 REM INIT SCREEN
110 REM -----------
```

```
120 COLOR 1,14,14:SCREEN 1:CLS
125 VPOKE BASE(6)+25,&HF4
135 VPOKE BASE(6)+6,&H13
137 LOCATE 5,0:PRINT"―――――――――――――――――
    ――――――――――"
140 LOCATE 5,1:PRINT"♬▲SOUND▲-▲M
    ASTER▲♬"
150 LOCATE 5,2:PRINT"―――――――――――――――――
    ――――――――――"
155  GOSUB 1000:REM PLAY SAMPLE
160 DIM LX(13),LY(13),S$(13)
170 DIM MV(13),IV(13),F$(13)
180 FOR X=0 TO 13
190 READ LX(X),LY(X),S$(X),MV(X)
    ,IV(X),F$(X)
200 LOCATE LX(X),LY(X):PRINT S$(
    X):LOCATE LX(X)+6,LY(X):GOSU
    B 5000
205 SOUND X,IV(X)
210 NEXT
220 LOCATE 23,12:PRINT"RRRTTT"
230 LOCATE 23,14:PRINT"CBACBA"
```

```
300 REM ACTIE
```

```
310 X=0:LOCATE LX(X)-1,LY(X):PRI
    NTP$
400 A=STICK(0):IF A=0 THEN 400
410 IF A=1 OR A=3 OR A=5 OR A=7
    THEN 450
420 GOTO 400
```



```
450 ON A GOSUB 500,1,600,1,700,1
    ,800
455 SOUND X,IV(X)
457 FOR DL=1 TO 200:NEXT
460 GOTO 400

500 REM OP

510 IF IV(X)<MV(X) THEN IV(X)=IV
    (X)+1:LOCATE LX(X)+6,LY(X):G
    OSUB 5000
520 RETURN

600 REM RECHTS

610 IF X= 13 THEN RETURN
620 LOCATE LX(X)-1,LY(X):PRINT"▲
    "
630 X=X+1:LOCATE LX(X)-1,LY(X):P
    RINT P$
635 FOR DL=1 TO 200:NEXT
640 RETURN

700 REM NEER

710 IF IV(X)>0 THEN IV(X)=IV(X)-
    1:LOCATE LX(X)+6,LY(X):GOSUB
     5000
720 RETURN

800 REM LINKS

810 IF X=0 THEN RETURN
820 LOCATE LX(X)-1,LY(X):PRINT"▲
    "
830 X=X-1:LOCATE LX(X)-1,LY(X):P
    RINTP$
835 FOR DL=1 TO 200:NEXT
840 RETURN
999 GOTO 999
1000 PLAY"O5V15S1M1000T255"
1010 PLAY"L8D+DL4DL8D+DL4DL8D+DL
     4DA+"
1020 PLAY"R4L8A+AL4GL8GFL4D+L8D+
     DL4CC"
1030 PLAY"R4L8D+DL4CL8DCL4CL8DCL
     4CAR4"
1040 PLAY"L8AGL4F+L8F+D+L4DL4F+G
     R2"
1050 PLAY"L8AGL4F+L8F+D+L4DL4F+G
     R2"
1055 IF PLAY(0)=-1 THEN 1055
1060 RETURN
2000 DATA 1,4,"CHA1",255,62,D
2010 DATA 15,4,"CHA2",15,0,D
2020 DATA 1,7,"CHB1",255,0,D
2030 DATA 15,7,"CHB2",15,0,D
2040 DATA 1,10,"CHC1",255,0,D
2050 DATA 15,10,"CHC2",15,0,D
2060 DATA 1,13,"NOISE",31,0,D
2070 DATA 15,13,"KAN.",63,62,B
2080 DATA 1,16,"VOLA",16,15,D
2090 DATA 15,16,"VOLB",16,0,D
2100 DATA 1,19,"VOLC",16,0,D
2110 DATA 15,19,"VORL1",255,50,D
2120 DATA 1,22,"VORL2",255,50,D
2130 DATA 15,22,"VORMN",15,13,D
5000 IF F$(X)="D" THEN PRINT USI
     NG "###";IV(X) ELSE PRINT U
     SING "########";VAL(BIN$(IV
     (X)))
5010 RETURN
```

	7	6	5	4	3	2	1	0 Bit
Register 0	Tone number channel A							
Register 1	0	0	0	0	Coarse tone A			
Register 2	Tone number B							
Register 3	0	0	0	0	Coarse tone B			
Register 4	Tone number channel C							
Register 5	0	0	0	0	Coarse tone C			
Register 6	0	0	0	Noise tone				
Register 7	0	0	RC	RB	RA	TC	TB	TA
Register 8	0	0	0	M	Volume A			
Register 9	0	0	0	M	Volume B			
Register 10	0	0	0	M	Volume C			
Register 11	Form-length tone							
Register 12	Form-length coarse tone							
Register 13	0	0	0	0	Form number			

Computer-adventure




```
10 COLOR11,1,1:CLS:SCREEN2:OPEN"GRP:" AS #1
20 FORX=180TO0STEP-10
30 Y=180-X
40 LINE(X,0)-(0,Y)
50 NEXTX
60 DRAW"BM47,50":COLOR11:PRINT#1,"HOLIDAY▲ON▲COMPUTER-ISLAND"
70 DRAW"BM30,70":PRINT#1,"(c)▲BY▲WIM▲DEWIJNGAERT▲(WDW)"
80 DRAW"BM30,80":PRINT#1,"▲▲▲▲▲▲▲IN▲APRIL▲1985"
90 FORT=1TO2000:NEXTT:CLOSE
100 SCREEN0:WIDTH40:KEYOFF:COLOR1,7
110 GOSUB 610
120 DIM Q$(50),G$(50),S%(50,4),V$(4),B%(60),N$(60),X$(30),N%(60)
130 X$(30)="Zelfs▲Superman▲is▲niet▲sterk▲genoeg▲om▲▲meer▲te▲dragen!"
140 A$="":P%=1:RESTORE:GOSUB770:GOSUB860
150 CLS
160 PRINTCHR$(7):PRINT"Ik▲ben▲:-":PRINTQ$(P%):A$="":A=FRE(""):IFS%(P%,1)>0THENA$="Noordelijk"
170 GOSUB770
180 IFS%(P%,3)>0ANDLEN(A$)>0THENA$=A$+",Oostelijk"ELSEIFS%(P%,3)>0THENA$="Oostelijk"
190 IFS%(P%,2)>0ANDLEN(A$)>0THENA$=A$+",Zuidelijk"ELSEIFS%(P%,2)>0THENA$="Zuidelijk"
200 IFS%(P%,4)>0ANDLEN(A$)>0THENA$=A$+",Westelijk"ELSEIFS%(P%,4)>0THENA$="Westelijk"
210 IFA$=""THENA$="niets▲voor▲de▲hand▲liggend!"
220 IFA%=8THENCLS:LOCATE10,2:PRINT"G▲O▲E▲D▲▲G▲E▲D▲A▲A▲N":PRINT:PRINT:PRINT:PRINT"Je▲hebt▲alle▲8▲onderdelen▲van▲de▲▲▲▲▲▲▲▲computerwereld▲uitgewisseld▲en▲verzameldvoor▲je▲reis▲naar▲huis!":END
230 PRINT:PRINT"Ik▲kan▲:-":PRINTA$
240 E=0:FORT=1TO19:PP%=0:IFB%(T)=P%THENPP%=1
250 IFPP%=1THEN270
260 NEXT:GOTO290
270 IFE=0ANDG$(T)<>""THENPRINT:PRINT"Ik▲zie▲▲:-"
280 PRINTG$(T):E=E+1:GOTO260
```

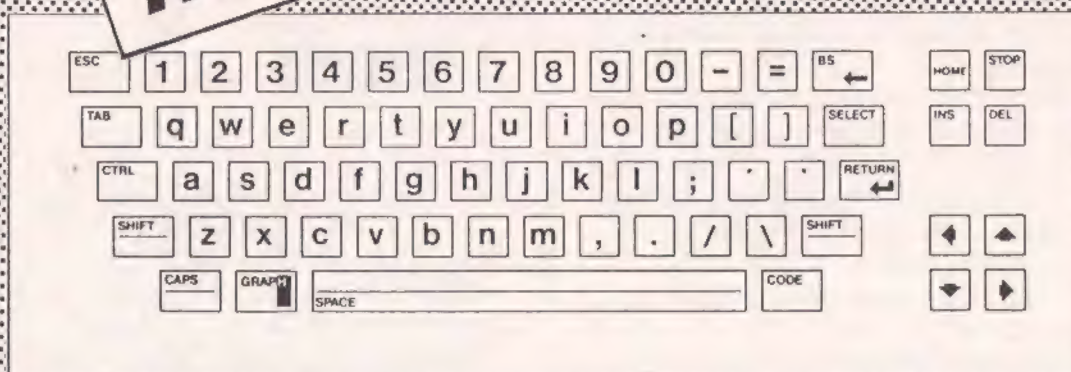
```
290 PRINT:PRINT"Wat▲moet▲ik▲nu▲d
    oen▲";:INPUT Z$:B$=LEFT$(Z$,
    2):C$=LEFT$(Z$,3):D$=LEFT$(Z
    $,4):E$=LEFT$(Z$,5):CLS:
300 IFE$="ga▲bi"ORE$="binne"ANDP
    %=20THENP%=21:GOTO160
310 IFE$="ga▲bi"ORE$="binne"THEN
    PRINT"Dat▲kan▲ik▲niet!":GOTO
    160
320 IFE$="ga▲bu"ORE$="buite"ANDP
    %=21THENP%=20:GOTO160
330 IFE$="ga▲bu"ORE$="buite"THEN
    PRINT"Dat▲kan▲ik▲hier▲niet!"
    :GOTO160
340 IFB$="ga"THENPRINT"Gebruik▲a
    lleen▲de▲richting▲N,O,Z,W!":
    GOTO160
350 IFC$="soo"THENGOSUB770:PRINT
    "Je▲hebt▲al▲";A%;"geldstukke
    n▲en▲je▲hebt▲er▲8▲nodig!":GO
    TO160
360 IFB$="n"ANDS%(P%,1)>0THENP%=
    S%(P%,1):GOTO160
370 IFB$="z"ANDS%(P%,2)>0THENP%=
    S%(P%,2):GOTO160
380 IFB$="o"ANDS%(P%,3)<>0ANDP%<
    >17THENP%=S%(P%,3):GOTO160
390 IFB$="o"ANDP%=17ANDAB<>1THEN
    PRINT"De▲hond▲wil▲me▲niet▲do
    or▲laten!":GOTO160
400 IFB$="o"ANDP%=17THENP%=S%(P%
    ,3):GOTO160
410 IFB$="w"ANDS%(P%,4)>0THENP%=
    S%(P%,4):GOTO160
420 IFB$="n"ORB$="o"ORB$="z"ORB$
    ="w"THENPRINT"Langs▲daar▲kan
    ▲ik▲niet▲gaan!":GOTO160
430 IFC$="gek"ORC$="zot"THENPRIN
    T"Wees▲niet▲onbeleeft!":GOTO
    160
440 IFC$="haa"ORC$="nee"ORC$="pa
    k"THENGOSUB880:GOTO160
450 IFC$="val"ORD$="laat"ORC$="z
    et"THEN1030
460 IFC$="inv"THENGOSUB1110:GOTO
    160ELSEIFC$="zwe"THEN1140
470 IFD$="klim"ANDP%=10THENP%=22
    :PRINT"O.K.":GOTO160ELSEIFD$
    ="klim"THENPRINT"Hier▲niet!"
    :GOTO160
480 IFD$="omho"ANDP%=10THENP%=22
    :GOTO160
490 IFC$="spr"ANDP%=22THEN1150EL
    SEIFC$="spr"THENPRINT"O.K.▲I
    k▲kom▲niet▲heel▲ver!":GOTO16
    0
500 IFD$="bene"ANDP%=22THENP%=10
    :PRINT"O.K.":GOTO160:ELSEIFD
    $="bene"THENPRINT"Doe▲niet▲z
    o▲gek!":GOTO160
510 IFD$="kijk"THEN160ELSEIFD$="
    werp"THEN1190
520 IFC$="eet"THENPRINT"Op▲dit▲o
    genblik▲heb▲ik▲geen▲honger!▲
    ▲▲▲▲Dank▲U":GOTO160
530 IFC$="dri"THENPRINT"Hier▲is▲
    niets▲om▲te▲drinken!":GOTO16
    0
540 IFC$="hel"THENPRINT"Dit▲kan▲
    ik▲niet▲oplossen,weet▲je!":G
550 IFD$="onde"THENPRINT"Ik▲zie▲
    niets▲biezonders!":GOTO160
560 IFD$="zoek"THEN1260
570 IFC$="gra"ANDP%=23THEN1290EL
    SEIFC$="gra"THENPRINT"Hier▲n
    iet!":GOTO160
580 IFD$="geef"THEN1310
590 PRINT"Sorry,maar▲ik▲begrijp▲
    U▲niet!":GOTO160
600 END
610 CLS:LOCATE7,2:PRINT"HOLIDAY▲
    ON▲COMPUTER-ISLAND":PRINT:PR
    INT:PRINT"Help▲mij▲A.U.B.!▲I
    k▲ben▲op▲vakantie▲▲▲▲▲en▲ik▲
    heb▲geen▲rooie▲duit▲meer.Ik▲
    moet▲▲ACHT▲dingen▲zoeken▲die
    ▲met▲de▲computer▲▲te▲maken▲h
    ebben▲en▲ze▲meenemen▲naar▲de
    "
620 PRINT"MSX-winkel▲om▲mijn▲sch
    atten▲uit▲te▲▲▲▲▲▲wisselen▲i
    n▲geld▲voor▲mijn▲terugreis."
630 PRINT:PRINT:PRINT"Je▲moet▲mi
    j▲instructies▲geven▲in▲klein
    e▲letters▲en▲in▲een▲of▲twee▲
    woordige▲▲▲▲▲▲zinnen▲(B.V.▲p
    ak▲eten▲of▲drink)"
640 PRINT:PRINT"DRUK▲EENDER▲WELK
    E▲TOETS:"
650 K$=INKEY$:IFK$=""THENGOTO650
    ELSERETURN
660 DATA op een smal voetpad met
     bomen aan elke  kant.,0,0,2
    ,0,aan het strand van een ko
    ud meer.,0,3,0,1,op een berg
    pad.,2,4,0,0,op de top van e
    en berg.Ik kan vier km.  zie
    n.,3,0,5,6
670 DATA bij een waterval.,0,10,
    0,4,op een smal bergpad.Het
    is erg mistig   hier.,0,7,4,
    0,bij een afgrond.Er zijn ov
    eral grote    keien.,6,8,0,0
680 DATA bij een grot.,7,0,9,0,i
    n een reusachtige grot.Er zi
    jn stalag- tieten en stalagm
    ieten overal.,0,0,11,8,bij d
    e waterval.Er is een zeer sm
    alle   boom hier.,5,11,0,0
```

```
690 DATA in de waterval.Ik wordt
     erg nat.Er is   een grotten
    -ingang hier.,10,0,12,9,aan
    de andere kant van de waterv
    al.,0,13,0,11,op een voetpad
    .,12,14,0,0
700 DATA aan de voet van een ber
    g.Er is een      klein dorpj
    e hier.,13,16,15,17,op een h
    oofdweg.,18,0,0,14,buiten he
    t dorpje is er een smederij.
    Hetis gesloten,14,0,0,0,bij
    enkele huizen.Er is hier een
     oude   dame.,0,0,14,0
710 DATA op een brug.,0,15,19,0,
    in de speeltuin,20,0,0,18,bi
    j de MSX-winkel.,0,19,0,0,bi
    nnen in de MSX-winkel.Er sta
    at een    bordje met 'Zet je
     onderdelen hier neer.Stuk p
    er stuk'.,0,0,0,0
720 DATA in een boom.,0,0,0,0,op
     een klein eiland in het mid
    den van   een groot meer.,0,
    0,0,0
730 DATA een lege bierkan,12,enk
    ele algen,11,een magere jong
    en,4,"",7,een ** GEHEUGEN **
    ,9,een ** PROGRAMMA **,16,ee
    n ** PRINTER **,17
740 DATA "",23,een ** DISK-DRIVE
     **,23,een waardevolle ** OP
    BERGKIST **,22,kristallen,11
    ,stenen,7,een eekhoorn,13,no
    ten,1,een schop,23,een hamer
    ,16,een boosaardige hond,17,
    een beetje eten voor de hond
    ,1
750 DATA een grote rolsteen,8,bi
    erkan,1,kan,1,algen,2,jongen
    ,3,zwitsers,3,paswoord,4,woo
    rd,4,geheugen,5,geheugen,5,p
    rogramma,6,programma,6,print
    er,7,rol,8,papier,8,disk,9,d
    rive,9
760 DATA opbergkist,10,kist,10,k
    ristallen,11,stenen,12,eekho
    orn,13,noten,14,schop,15,ham
    er,16,hond,1,eten,18,rolstee
    n,19
770 A%=0:IFB%(3)=21THENA%=A%+1
780 IFB%(4)=21THENA%=A%+1
790 IFB%(5)=21THENA%=A%+1
800 IFB%(6)=21THENA%=A%+1
810 IFB%(7)=21THENA%=A%+1
820 IFB%(8)=21THENA%=A%+1
830 IFB%(9)=21THENA%=A%+1
840 IFB%(10)=21THENA%=A%+1
850 RETURN
860 FORX=1TO23:READQ$(X):FORY=1T
    O4:READS%(X,Y):NEXTY,X
870 FORX=1TO19:READG$(X),B%(X):N
    EXT:FORX=1TO27:READN$(X),N%(
    X):NEXT:RETURN
880 GOSUB980:IFL%=1THEN900
890 RETURN
900 E%=0:FORH=1TO19:IFB%(H)=P%AN
    DB%(N%(R))=P%THENE%=1
910 NEXT:IFE%=0THENRETURN
920 IFR=23THENAA=AA+1
930 IF(R=4ORR=5)ANDAC<>1THENPRIN
    T"Dat▲kan▲ik▲niet▲doen▲!":RE
    TURN
940 IFR=25THENPRINT"Het▲heeft▲me
    ▲woedend▲aangevallen▲▲▲▲▲▲▲▲
    AAGGGHHHH":GOTO1160
950 E%=0:FORD=1TO3:IFV$(D)=""THE
    NV$(D)=G$(N%(R)):E%=1:D=7
960 NEXT:IFE%=0THENPRINTX$(30):R
    ETURN
970 B%(N%(R))=0:RETURN
980 L$="":FORH=1TOLEN(Z$)
990 IFMID$(Z$,H,1)="▲"THENL$=RIG
    HT$(Z$,(LEN(Z$)-H)):H=90
1000 NEXT:R=0:L%=0:IFLEN(L$)<2TH
     ENRETURN
1010 FORH=1TO27:IFLEFT$(N$(H),LE
     N(L$))=L$THENL%=1:R=H
1020 NEXT:RETURN
1030 GOSUB980:IFL%=1THEN1050
1040 PRINT"Ik▲zie▲geen▲";L$:GOTO
     160
1050 E%=0:FORD=1TO3:IFV$(D)=G$(N
     %(R))THENV$(D)="":E%=1
1060 NEXT:IFE%=1THEN1080
1070 PRINT"Ik▲heb▲het▲niet▲eens!
     ":GOTO160
1080 B%(N%(R))=P%:IFR=23THENAA=0
1090 IFP%=17ANDR=26THENAB=1:PRIN
     T"Het▲is▲goed▲aan▲het▲eten.
     Ik▲denk▲dat▲ik▲kan▲ontsnapp
     en!"
1100 GOTO160
1110 PRINT"Ik▲sleep▲aan:-":PRINT
     V$(1):PRINTV$(2):PRINTV$(3)
1120 IFV$(1)=""ANDV$(2)=""ANDV$(
     3)=""THENPRINT"Helemaal▲nik
     s!":F%=0ELSEF%=1
1130 PRINT:RETURN
1140 IFP%=2THENP%=23:PRINT"O.K."
     :GOTO160ELSEIFP%=23THENP%=2
     :PRINT"O.K.":GOTO160ELSEPRI
     NT"Ik▲kan▲hier▲niet▲zwemmen
     ▲IDIOOT!":GOTO160
1150 CLS:LOCATE0,10:PRINT"▲▲Ik▲v
     iel▲en▲brak▲mijn▲nek▲!!!!!!
     !!!!!!!!"
```

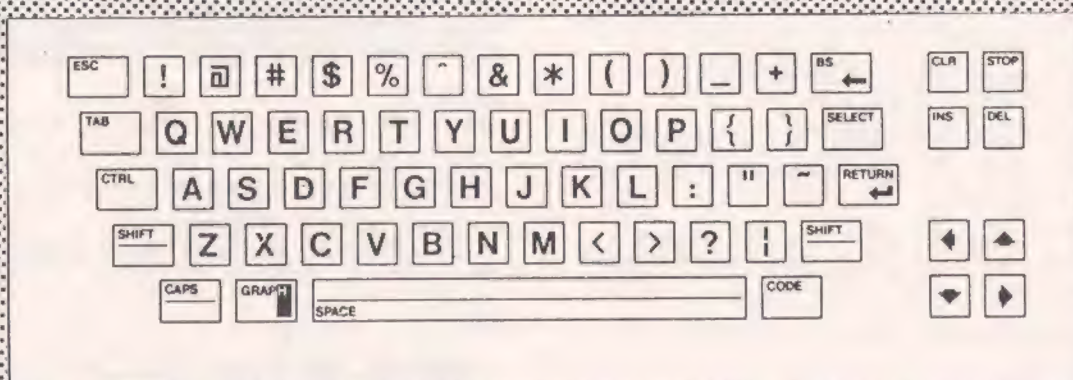
```
1160 PRINT:PRINT:PRINT"▲▲Ik▲ben▲
     dood.Druk▲SELECT▲om▲terug▲t
     e▲▲▲▲▲spelen."
1170 A$=INKEY$:IFA$=CHR$(24)THEN
     RUN
1180 IFA$="N"ORA$="n"THENPRINT"D
     aaag!!!":PRINT:PRINT:ENDELS
     EGOTO1170
1190 GOSUB980:IFL%=1THEN1210
1200 PRINT:PRINT"Ik▲zie▲geen";L$
     :GOTO160
1210 E%=0:FORD=1TO3:IFV$(D)=G$(N
     %(R))THEN V$(D)="":E%=1
1220 NEXT:IFE%=1THEN1240
1230 PRINT"Ik▲heb▲het▲niet▲gehad
     !":GOTO160
1240 IFP%<23THENB%(N%(R))=P%+1EL
     SEB%(N%(R))=P%
1250 PRINT"O.K.":GOTO160
1260 IFP%=7THEN1270ELSEPRINT"Ik▲
     kan▲niets▲vinden!":GOTO160
1270 G$(4)="een▲**▲PASWOORD▲**":
     IFAZ<>0THENPRINT"Er▲is▲niet
     s▲anders▲hier":GOTO160
1280 PRINT"Ik▲heb▲iets▲gevonden!
     ":AZ=1:GOTO160
1290 IFAA<>1THENPRINT"Ik▲heb▲nie
     ts▲om▲mee▲te▲graven!":GOTO1
     60
1300 PRINT"Ik▲heb▲iets▲gevonden!
     ":G$(8)="een▲rol▲**▲PAPIER▲
     **":AQ=2:GOTO160
1310 GOSUB980:IFL%=1THEN1330
1320 GOTO160
1330 IFP%=4ANDR=22THENPRINT"Hij▲
     nam▲een▲beetje▲en▲gaf▲me▲ie
     ts▲in▲▲▲▲ruil▲en▲liep▲dan▲z
     ingend▲weg.":G$(3)="een▲**▲
     TOETSENBORD▲**":N$(4)="toet
     senbord":AC=1:GOTO160
1340 IFP%=21ANDR=22THENPRINT"De▲
     eekhoorn▲lust▲ze▲niet!":GOT
     O160
1350 PRINT"Ik▲begrijp▲je▲niet▲go
     ed!":GOTO160
```

```
☺☻♥♦♣♠·◘○◙♂
♀♪♫☼┼┴┬┤├┼│─
┌┐└┘╳╱╲+ !"#
$%&'()*+,-./
0123456789:;
<=>?@ABCDEFG
HIJKLMNOPQRS
TUVWXYZ[\]^_
`abcdefghijk
```

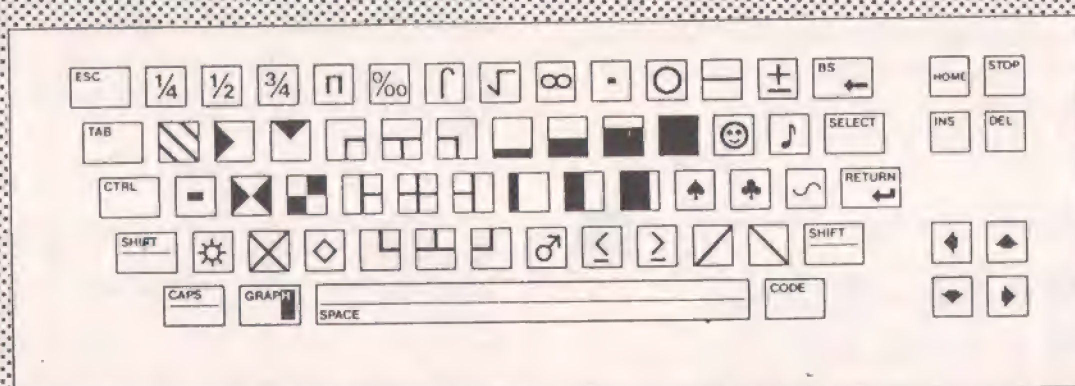
MSX Toetsenbord



A. STANDAARD

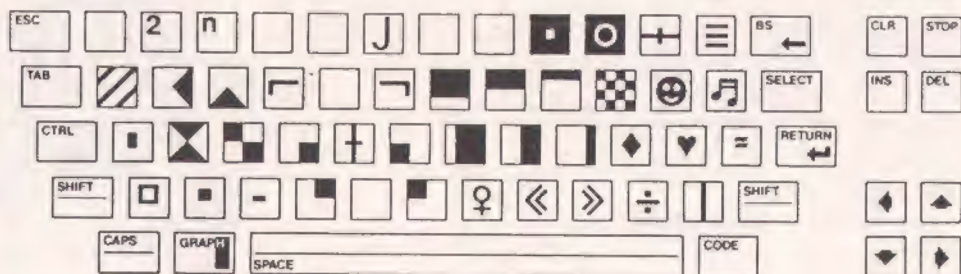


B. + SHIFT

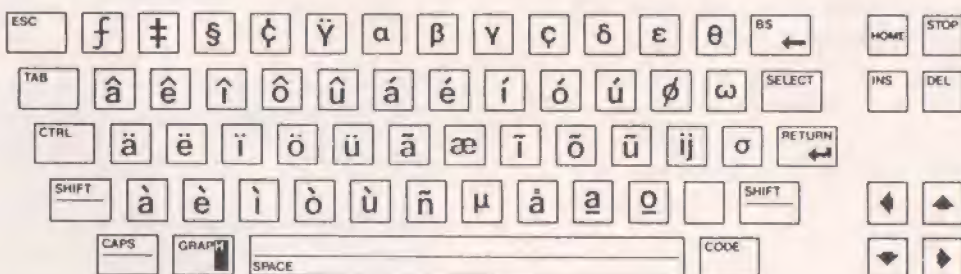


C. + GRAPH

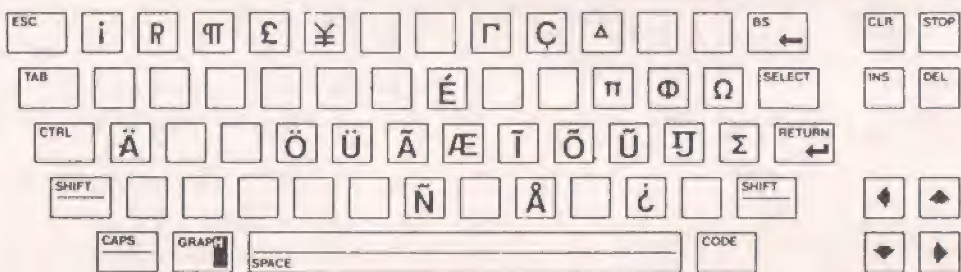
MSX Toetsenbord



D. GRAPH + SHIFT



E. + CODE



F. CODE + SHIFT

Spritemaker

SPRITEMAKER

Bij een eerste gebruik van dit programma kiest men voor een 8 x 8 of 16 x 16 formaat.
Dan verschijnt het **bord** waarop men met behulp van pijltjes de CURSUR kan laten bewegen.
Naast het **bord** verschijnt trouwens de werkelijke sprite.
Onderaan is de betekenis van de cursor aangeduid :

(M)ove
(P)ut
(E)rase.

Vanuit de huidige stand kan men ook het scherm (C)learen: volgt hier een code 11 dan wordt alleen de actuale sprite gewist.
Met 22 volgt een herstart met nieuwe formaatkeuze.
Vanuit deze positie kan men ook naar een andere situatie d.m.v. TAB.
Nu kan men elke sprite beheersen door zijn nummer op te (R)oepen men kan deze sprite een nieuwe (K)leur geven of hem ook (D)eponeren op een bepaalde positie.
Dit laatste houdt dus ook in dat men sprites kan verdubbelen, evt. in een andere kleur.
De pijltjes beheersen in dit geval ook de opgeroepen sprite, zodat men de afgebeelde sprites naast en onder elkaar kan schikken om zo tot een geheel te komen.
Van hieruit kan men ook het totaal der opgebouwde sprites (S)aven (in SCII-formaat).
Met (ESC) kunnen we naar het **bord** terugkeren.
De geSAVEde data kunen dan met hetzelfde of een ander programma worden gemerged.
Bij een volgend gebruik zal men trouwens beginnen mat na te gaan of aan het programma geen sprite-data zijn toegevoegd.
Het DELETEN van deze staat trouwens los van de CLEAR in het algemeen.

R.CEUTERICK
Smidsestraat 204 GENT




```
10 REM █████ Spritemaker ██████████
   ██              versie 1.0 - TRO
   N
20 REM █████ R.Ceuterick, Gent.
30 REM

40 CLEAR2000:DEFINT A-Z:DIMA$(32
   ),GG$(30)

50 REM

60 O$(0)="▲":O$(1)="█"
70 COLOR 1,3,3:SCREEN1:KEYOFF
80 PRINT"▲▲▲▲▲SPRITEMAKER.▲▲▲▲▲▲
   ▲▲▲▲▲▲▲▲▲▲▲▲───────────────────":PRIN
   T
90 PRINT"Het▲programma▲start▲met
   ▲de▲▲▲reeds▲opgenomen▲sprites
   ▲of▲▲▲wacht▲zoniet▲op▲recorde
   rin-▲▲voer.▲<RETURN>"
100 PRINT:PRINT"Tik▲(9)▲voor▲DEL
    ETE▲van▲reedsbestaande▲data.
    ▲Run▲opnieuw.":PRINT
110 PRINT:PRINT"Zoniet,▲kies▲je▲
    formaat:":PRINT
120 PRINT"▲▲▲▲▲▲▲▲▲(1)=8x8▲▲▲▲▲▲▲
    ▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲(2)=16x16":PR
    INT
```

```
130 F=VAL(INPUT$(1)):IFF<0 ORF>2
     THEN130
140 IFF=0 THEN960
```

```
150 REM ██ vastleggen formaat
```

```
160 SCREEN1,F:PUTSPRITE(0),(200,
    40),0:SK=1
170 IFF>0 THENF=8*F ELSE10
180 GOSUB470
190 FORZ=1TOF:A$(Z)=STRING$(F,48
    ):NEXT:F2=8
200 IFF>8 THENF2=32
210 X$=STRING$(F2,0)
```

```
220 REM ██ 'bord'- gebruik
```

```
230 H=4:V=4:H1=4:V1=4:I$="M":T=0
    :SN=0:GOSUB530:GOTO250
240 I$=INKEY$:IFI$=""THEN240
250 FF=INSTR("PEMCpemc",I$):ON F
    F GOSUB430,440,450,720,430,4
    40,450,720:FF=0
260 IFI$=CHR$(28) AND H<F THENH=
    H+1
270 IFI$=CHR$(29) AND H>1 THENH=
    H-1
280 IFI$=CHR$(30) AND V>1 THENV=
    V-1
290 IFI$=CHR$(31) AND V<F THENV=
    V+1
300 IFI$=CHR$(9)THENGG$=X$:SN=0:
    GOTO570
310 LOCATEH1,V1:PRINTO$(T)
```

```
320 REM ██ aanp. beeld&sprite
```

```
330 LOCATEH,V
340 IFCU$="M"THENU$=MID$(A$(V),H
    ,1):T=VAL(U$):GOTO360
350 MID$(A$(V),H,1)=G$
360 PRINTCU$:H1=H:V1=V
370 W1=VAL("&B"+LEFT$(A$(V),8))
380 W2=VAL("&B"+RIGHT$(A$(V),8))
390 MID$(X$,V,1)=CHR$(W1)
400 IF F>8 THENMID$(X$,V+16,1)=C
    HR$(W2)
410 SPRITE$(0)=X$:SN=0:GOSUB540
420 GOTO240
430 G$="1":CU$="P":T=1:RETURN
440 G$="0":CU$="E":T=0:RETURN
450 CU$="M":RETURN
```

```
460 REM ██ afb. 'bord'scherm
```

```
470 CLS:PRINT"▲";STRING$(F,"◇")
480 FORZ=1TOF:PRINT"◇";SPC(F);"◇
    ":NEXT
490 PRINT"▲";STRING$(F,"◇")
500 LOCATE0,19:PRINT"(P)ut,(E)ra
    se,(M)ove,▲<TAB>▲▲(C)lear";S
    PC(40)
510 RETURN
```

```
520 REM ██ sprites
```

```
530 PUTSPRITESN,STEP(0,0),SK
540 PUTSPRITESN,STEP(SH,SV)
550 RETURN
```

```
560 REM ██ manipulatie sprites
```

```
570 LOCATE0,19:PRINTUSING"Spr=##
    .▲";SN;:PRINT"(K)leur,(P)laa
    ts,▲▲▲▲<ESC>▲▲▲(R)oep▲,(S)av
    e"
580 I$=INKEY$:IFI$=""THEN580
590 IFINSTR("KPRSkprs",I$)>0THEN
    660
600 IFI$=CHR$(27)THENGOSUB500:GO
    TO240
610 IFI$=CHR$(28)THENSH=1
620 IFI$=CHR$(29)THENSH=-1
630 IFI$=CHR$(30)THENSV=-1
640 IFI$=CHR$(31)THENSV=1
650 GOSUB540:SH=0:SV=0:GOTO570
660 IFI$="S" OR I$="s"THENGOTO78
    0
670 CO$=I$+"=":GOSUB750
680 IF(I$="k"ORI$="K")AND NR>0 A
    ND NR<16 THENSK=NR:GOSUB530
690 IF(I$="P" OR I$="p") ANDNR>0
     AND NR<30 THENSN=NR:SPRITE$
    (SN)=GG$:GG$(SN)=GG$:GOSUB53
    0:GOTO570
700 IF(I$="R" OR I$="r") ANDNR>=
    0 AND NR<30 THENSN=NR:GG$=GG
    $(SN)
710 GOTO570
720 CO$="CLEAR?▲(11)=sor▲(22)=tt
    l":GOSUB750
730 IFNR=11THEN180
740 IFNR=22THEN40 ELSE RETURN
750 LOCATE0,21:PRINTSPC(50):LOCA
    TE0,21:PRINTCO$;
760 INPUTAN$:NR=VAL(AN$):LOCATE0
    ,21:PRINTSPC(50):RETURN
```

```
770 REM ■ uitvoer
```

```
780 LOCATE0,19:PRINT"▲RECORDER▲KLAAR?▲Bevestig▲S,▲▲of▲<M>otor?▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲"
790 I$=INPUT$(1)
800 IFI$="m"ORI$="M"THEN MOTOR:GOTO780
810 IFI$<>"S"THEN570
820 OPEN"cas:spr$'s" FOR OUTPUT AS 1
830 Z=0:PRINT#1,"60000REM▲F2=32▲voor▲16x16,▲▲▲▲▲▲▲▲▲▲▲▲zoniet▲8";CHR$(13)
840 PRINT#1,"60001DATA▲";F2;CHR$(13)
850 Z=Z+1:IFGG$(Z)=""THEN890
860 PRINT#1,60000!+10*Z;"DATA▲";
870 FORY=1TOF2:W=ASC(MID$(GG$(Z),Y)):PRINT#1,W;:IFY<F2THENPRINT#1,",";
880 NEXT:PRINT#1,CHR$(13)
890 IFZ<30THEN850
900 PRINT#1,"61990end";CHR$(13);"62000delete60000-62000";CHR$(13)
910 CLOSE:MOTOR ON:FORZ=1TO400:NEXT:MOTOROFF:GOTO180
```

```
920 REM ■ invoer
```

```
930 SCREEN1:PRINT:PRINT"▲▲▲▲RECORDER▲loopt▲▲▲▲▲▲▲▲▲▲▲▲voor▲LADEN▲van▲SPRITES."
940 MERGE"cas:spr$'s"
```

```
950 REM ■ omz. data-sprites
```

```
960 ONERROR GOTO1080
970 RESTORE60000:READF2:F=1:IFF2>8 THENF=2
980 Z=1
990 ON ERRORGOTO1080
1000 GG$(Z)="":FORY=1TOF2:READQ:GG$(Z)=GG$(Z)+CHR$(Q):NEXT
1010 Z=Z+1:GOTO1000
1020 SCREEN1,F:GS$=""
1030 FORZ=1TO30:SPRITE$(Z)=(GG$(Z)):IFLEN(GG$(Z))>7THENGS$=GS$+"▲"+STR$(Z)
1040 SK=RND(1)*10+3:PUTSPRITEZ,(240-8*Z,6*Z),SK
1050 NEXT:PUTSPRITE0,(220,90)
1060 CLS:LOCATE0,14:PRINTGS$
1070 GOTO170
1080 IFERR=8THENRESUME930
1090 IFERR=4 THENRESUME1020
1100 END
```

SPRITEMAKER.

Het programma start met de reeds opgenomen sprites of wacht zoniet op recorderinvoer. <RETURN>

Tik (9) voor DELETE van reeds bestaande data. Run opnieuw.

Zoniet, kies je formaat:

(1)=8x8
(2)=16x16



(P)ut,(E)rase,(M)ove, <TAB>
(C)lear



(P)ut,(E)rase,(M)ove, <TAB>
(C)lear



Europees Computer Centrum

Dr. Persoonslaan 15b
Willebroek
Telefoon 03 - 886 2060



MSX Expansion Computer Case

	Hfl	BF
ECC 8	550,--	9.900
ECC 4	375,--	6.750
Goldstar MSX 64k	610,--	10.999
AVT MSX 64k	765,--	13.990
Deawoo kleurenmonitor	910,--	16.380

Lees de objectieve testrapporten in:
- MSX Magazine
- MSX Info
- MSX Mozaïek
- MSX Verenigingsblad België
- Radio Amateur Magazine

MSX



	Hfl	BF
eerste diskdrive 5,25 inch		
eerste diskdrive 180 Kb	995,--	17.100
eerste diskdrive 360 Kb	1638,--	29.485
eerste diskdrive 720 Kb	1679,--	30.220

Tweede diskdrive mag 5,25 of 3,5 inch zijn. In kast met voeding en alle bijbehorende kabels. Bij de genoemde prijzen zijn wij ervan uitgegaan dat U de eerste drive al van ons in Uw bezit heeft.

	Hfl	BF
tweede diskdrive 180 Kb	950,--	17.100
tweede diskdrive 360 Kb	1056,--	19.000
tweede diskdrive 720 Kb	1095,--	19.710



EPROM PROGRAMMER
VOOR MSX COMPUTERS inclusief software

Hiermee kan U Uw zelfgemaakte basic (of machinetaal) programma's in eprom zetten (ook bestaande software).

Hfl	BF
499,--	8.980

Rompacks met eproms los verkrijgbaar.
2K-4K-8K en 16K.

Volwaardige MSX printers door middel van de door ons vervaardigde Rompack printer routine.

Indien U een FAX 180 of Trend JP printer bezit, kunt U nu alle MSX karakters uitprinten.

Vraag de introductieprijs! ! ! met of zonder printer.

Verder verkopen wij MSX computers, monitoren, printers en alle toebehoren. Nieuw is de MSX MUIS. Natuurlijk ook de door onszelf ontwikkelde Expansion sloten, ECC 8 en ECC 4. Tegen zeer scherpe prijzen.

Prijzen inclusief BTW. Prijswijzigingen voorbehouden.
Telefoonnummer van ECC vanuit Nederland 09 - 323 886 2060.
Wij zijn maandag tot vrijdag van 09.00 tot 18.00 uur en zaterdag van 10.00 tot 18.00 uur open.
Voor schriftelijke reactie: Postbus 73817 te Den Haag.

DEALERS VOOR DE DOOR ONSZELF ONTWIKKELDE HARDWARE WELKOM



MSX AVAILABLE NOW ON ROM CARTRIDGE MSX

Track and Field I
Track and Field II
Hyper Sports I
Hyper Sports II

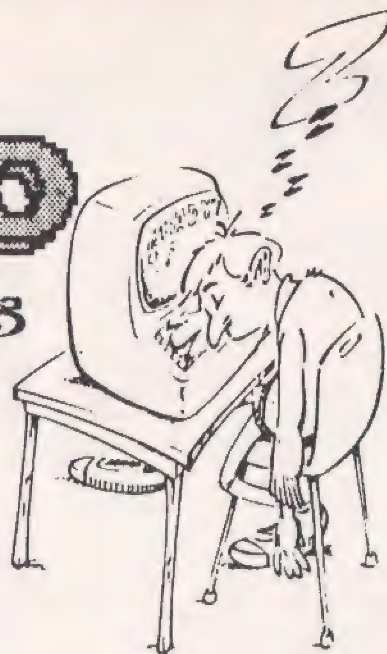
Konami's Tennis
Time Pilot
Super Cobra
Athletic Land

Antarctic Adventure
Comic Bakery
Monkey Academy
Circus Charlie

Konami®

KONAMI LTD., TELEVISION HOUSE, 269 FIELD END RD, EASTCOTE, MIDDLESEX HA4 9L
ENGLAND. TEL: (01) 429 2446 TLX 883978 KONAMI G

Troninfo ONDERWIJS





NOTENLEER

BEDOELING VAN HET PROGRAMMA:

In deze programma's worden enkele lessen geprogrammeerde instruktie gepresenteerd rond notenleer. De kennis van de toonladder, tempo, toonhoogte e.d. staan hierbij centraal.

OMSCHRIJVING :

De geprogrammeerde instruktie bestaat uit 5 lessen die de leerlingen moeten doorlopen. Telkens verschijnt een tekst geillustreerd met enkele tekeningen. Ook worden de lesjes wat opgefleurd door allerlei beeptoontjes die in computerland moeten doorgaan voor musikale klanken.
Na een instruktie-schakel wordt aan de gebruiker een vraagje gesteld met betrekking tot de geleerde informatie.

WE VONDEN POSITIEF

* de wijze waarop de lessen worden geillustreerd

* de stijgende moeilijkheidsgraad

WE VONDEN MINDER GOED

* de talrijke TAALFOUTEN die in bijna elke instruktie-schakel te vinden zijn

* de opeenhoping van de leesregels wat het lezen bemoeilijkt.

* het gebrek aan een degelijke foutenanalyse (de gebruiker moet opnieuw de instruktie-schakel doornemen)

WAARDERING (op 10) : 5

LESSEN IN BASIC

BEDOELING VAN HET PROGRAMMA

Via 14 lessen wordt met deze cassette gepoogd aan de lesvolger een basis te geven in de computertaal "BASIC".

DE WERKING VAN HET PROGRAMMA

De programma-reeks is van het type van de "geprogrammeerde instruktie". Via een aantal rijk geillustreerde lesjes worden de belangrijkste instrukties van de "MSX"-basic op een rijtje geplaatst.
Regelmatig worden enkele vraagjes gesteld om de opgedane kennis te controleren.
De regelnummers van korte programma's worden een voor een verklaard.

WIJ VONDEN POSITIEF :

* de opbouw van de verschillende lessen

* de prettige illustraties bij elke instruktie-schakel

WIJ VONDEN MINDER GOED :

* de talrijke taalonzuiverheden

WAARDERING (op 10) : 7



LOGO van Kuma

BEDOELING VAN HET PROGRAMMA

LOGO is een computertaaltje ontwikkeld voor kinderen. Het leren logisch redeneren en het meetkundig inzicht vormen de belangrijkste doelstellingen van deze taal. Wij bespreken dit programma hoofdzakelijk voor deze mensen die reeds een voorafgaandelijke kennis bezitten van de LOGO-taal.

WERKING VAN HET PROGRAMMA

Nadat het programma werd ingeladen verschijnt de titel-pagina, gevolgd door een grafisch scherm waarop in de linker bovenhoek een kader verschijnt en in het midden een figuurtje vooruit schuift dat moet doorgaan voor een schildpad. Men kan rechtstreekse instrukties intijpen (deze verschijnen in het kadertje) of een procedure opbouwen (d.w.z. een nieuwe instruktie aanleren). In het eerste geval beschikt men over een aantal woordjes zoals DRAW, FORWARD, LEFT, RIGHT, JUMP, CIRCLE, COLOUR, ERASE, PENUP, PENDOWN enz... terwijl in het andere geval men een nieuw woordje moet voegen bij het repertorium van de LOGO-taal bvb TO DRIEHOEK

```
TO DRIEHOEK
   DRAW 30
   LEFT 120
   DRAW 30
   LEFT 120
   DRAW 30
   END
```

laat toe dat de schildpad een driehoek tekent met 30 stappen voor elke zijde.

WIJ VONDEN POSITIEF :

* Enkele nieuwe instrukties die in de LOGO-taal werden opgenomen (CIRCLE, DISC...)

WIJ VONDEN MINDER GOED :

* de beperkte woordenschat van deze LOGO-taal

* het feit dat de taal in BASIC is geschreven

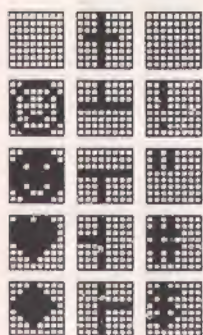
* de programmeer-fouten welke maken dat de lijnen niet recht worden getekend (NB degenen die dit programma gekocht hebben kunnen dit verbeteren door bij elke SIN- en COS-formule de instruktie FIX te voegen)

* de gebrekkige INPUT-routine die verbeteringen onmogelijk maakt en onzichtbare fouten in de hand werkt

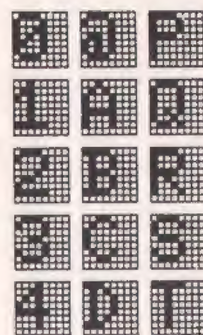
* de blokeringen tijdens het programma (te wijten aan herschikkingen in de reeks-variabelen)

WAARDERING (op 10) : 4

Videodisplay deel 2



**De vorige keer hebben we gezien hoe de tekstschermen van een MSX-computer opgebouwd zijn en wat we er mee kunnen doen.
Ditmaal gaan we kijken hoe het grafisch scherm (SCREEN 2) is opgebouwd. We gaan tevens de Basic-opdrachten die gebruik maken van dit grafisch scherm toelichten.
In een derde deel van deze reeks komt het tweede grafisch scherm (SCREEN 3) aan bod samen met (heel uitgebreid) de fameuze SPRITES.**



2.1.Screen 2.

Hier hebben we 192 lijnen van 256 punten. De plaats van de punten zijn bepaald door twee coördinaten op te geven,een X-coördinaat voor de plaats op de lijn, een Y-coördinaat voor de lijn waarop het punt zich bevindt. De oorsprong van het assenstelsel dat we zo bekomen bevindt zich links-boven. We werken dus in het tweede kwadrant, met het verschil dat we geen negatieve getallen gebruiken.
Ieder punt kan met de Basic opdrachten een bepaalde kleur gegeven worden. We moeten er wel op letten dat we, in een groep van acht punten die zich op dezelfde lijn bevinden en aan mekaar grenzen, maar twee verschillende kleuren kunnen gebruiken. Het waarom hiervan wordt duidelijk als we gaan kijken in welke delen het Video-Ram is opgesplist.

2.2. Een namentabel.

Voor elke matrix van 8x8 op het beeldscherm hebben we hier een geheugenplaats. Zie figuur 7. Ze is net als bij SCREEN 1 768 bytes lang. Hierin staan de waarden die bepalen welk patroon uit de patroontabel moet genomen worden om op de plaats van het karakter af te beelden. Het begin ervan vinden we in de speciale variable BASE(10).

2.3. Een patroontabel.

In tegenstelling met SCREEN 0 en 1 kunnen we in deze patroontabel voor elke matrix van 8x8 op het beeldscherm een uniek patroon samenstellen. De tabel is daarvoor 768x8=6144 bytes lang. De acht bytes van ieder patroon worden onder mekaar op het scherm afgebeeld. Het begin ervan vinden we in de speciale variable BASE(12).
Tik het volgende programma in en RUN het.

```
10 COLOR 15,4,15
20 SCREEN 2
30 VPOKE BASE(12)+32,&B10101010
40 GOTO 40
```

Met regel 10 stellen we kleur van het scherm in, wit (15) als voorgrond, blauw (4) als achtergrond en weer wit voor de rand. Met regel 20 schakelen we over op het grafisch scherm. Met regel 30 brengen we een patroon aan in de eerste byte van het vierde patroon. Het zal daardoor zichtbaar worden op de eerste lijn van het scherm. Met regel 40 verhinderen we het terug overschakelen op de tekststand.

2.4. Naam en patroontabel samen.

Vermits de patroontabel een lengte heeft voor 768 unieke patronen is deze, samen met de naamtabel, in drie verdeeld. Elk deel van de naamtabel kan alleen wijzen naar zijn deel van de patroontabel. Deze verdeling is nodig omdat een byte maximum 256 verschillende patronen kan aanwijzen. Bij het initialiseren van SCREEN 2 wordt elk byte van de naamtabel gevuld met een waarde opklimmend vanaf 0, zodat de eerste byte naar het eerste patroon wijst, de tweede byte naar het tweede patroon enz.
Tik volgende regel bij in :

```
35 VPOKE BASE(10)+10,4
```

Hiermee laten we de tiende byte in het eerste deel van de naamtabel eveneens naar het vierde patroon in de patroontabel wijzen. Dit heeft voor gevolg dat het bitpatroon tweemaal op de eerste lijn van het scherm zal verschijnen.

2.5. Een kleurentabel.

Deze tabel is even lang als de patroontabel. De informatie die erin vervat zit is identiek als deze van SCREEN 1. We kunnen daardoor elke byte in de patroontabel maximum twee kleuren geven, een voorgrond en een achtergrondkleur. Bits die een zijn worden afgebeeld met de voorgrondkleur, bits die 0 zijn met de achtergrondkleur. Vermits elk patroon uit acht bytes bestaat is het mogelijk in een patroon de 16 kleuren weer te geven. Het begin van deze tabel vinden we in de speciale variable BASE(11).
Tik het volgende programma in :

```
10 COLOR 15,4,4
20 SCREEN 2
30 FOR C%=0 TO 15 STEP 2
40  VPOKE BASE(12)+32+I%,&B11110000
50  VPOKE BASE(11)+32+I%,C%*16+C%+1
60  I%=I%+1
70 NEXT C%
80 GOTO 80
```

Met regel 40 in de FOR-NEXT lus vullen we iedere byte van het vierde patroon met dezelfde informatie. Met regel 50 vullen we de overeenstemmende bytes in de kleurtabel met de 16 verschillende kleuren, de oneven kleuren als voorgrond (C%*16), de even kleuren (C%+1) als achtergrond. I% fungeert als teller in de patroon en kleurtabel.

3. De grafische BASIC opdrachten.

Nadat het video-ram is geinitialiseerd voor SCREEN 2 is de namentabel gevuld met waarden, zodanig, dat elke byte ervan naar z'n eigen patroon in de patroontabel wijst. Elke bit van elke byte in de patroontabel wordt op nul gezet, en de kleuren in de kleurtabel op de waarden welke gebruikt werden in het laatste COLOR statement. Alles is nu klaar om de grafieken en figuren te ontvangen die we met de basic opdrachten gaan maken.

3.1. PSET en PRESET.

Met deze opdrachten kunnen we, met PSET, een punt in een voorgrondkleur en, met PRESET, in een achtergrondkleur kleuren.
Tik het volgende programma in :

```
10  SCREEN 2
20  REM DEEL 1
30  FOR X=0 TO 255 STEP 10
40   PSET (X,0)
50   FOR Y=0 TO 191 STEP 10
60    PSET STEP(0,10)
70   NEXT Y
80  NEXT X
90  REM DEEL 2
100  FOR X=0 TO 255 STEP 10
110   PRESET (X,0)
120  FOR Y=0 TO 191 STEP 10
130   PRESET STEP(0,10)
140  NEXT Y
150 NEXT X
160 GOTO 20
```

Het eeste deel van dit programma zet een reeks punten, met een onderlinge afstand van 9 punten, op het scherm in de voorgrondkleur. Het tweede deel zet dezelfde punten terug in de achtergrond kleur. Regel 160 zorgt voor een herhaling van het geheel.
In regel 40 en 110 zorgen de statements voor de exacte coordinaten van het te kleuren punt, terwijl in regel 60 en 130, met toevoeging van de functie STEP, de coordinaten relatief zijn ten opzichte van de laatst gebruikte coordinaten.
Dit wil zeggen dat de X en Y waarden, binnen de haken, bij de vorige coordinaten opgeteld worden om de nieuwe coordinaten te verkrijgen. Dit houdt in dat relatieve coordinaten negatief kunnen zijn. De functie STEP is bruikbaar met elke grafische opdracht.
Voeg volgende regels toe :

```
25  FOR C=0 TO 15
135 NEXT C
```

Wijzig de volgende regels in :

```
30  PSET (X,0),C
50  PSET STEP(0,10),C
140 GOTO 25
```

Regel 15 en 135 zorgen, samen met de gewijzigde regels, ervoor dat de cyclus doorlopen wordt voor de 16 kleuren.
Het gebruiken van een kleur kan zowel gebeuren door PSET als PRESET, het verschil zit erin dat, voor PSET, de overeenkomstige bit in de patroontabel een wordt en voor PRESET nul. De overeenkomstige kleuren worden in de kleurtabel geplaatst.

3.2. LINE.

Deze opdracht gebruiken we om lijnen, rechthoeken en rechthoekige vlakken te tekenen. Om een lijn te tekenen moeten we twee coordinaten opgeven. Deze coordinaten kunnen, met toevoeging van STEP relatief gemaakt worden.
Tik het volgende programma in :

```
10 SCREEN 2
20 PSET (0,191)
30 FOR C=0 TO 15
40  LINE STEP(0,-191)-STEP(16,191),C
50 NEXT C
60 GOTO 60
```

Dit programma tekent 16 lijnen op het scherm in de 16 kleuren. De eerste lijn zal dezelfde kleur hebben als de rand omdat ze in kleür 0 (transparant) is getekend. Een van de andere lijnen zal dezelfde kleur hebben als de achtergrond. Wijzig de volgende regel in :

```
40  LINE STEP(0,-191)-STEP(16,191),C,B
```

Door toevoeging van de letter B geven we ditmaal opdracht om rechthoeken te tekenen.
Wijzig de volgende regel in :

```
40  LINE STEP(0,-191)-STEP(16,191),C,BF
```

Ditmaal geven we, door toevoeging van de letters BF de opdracht om rechthoekige vlakken te tekenen met als gevolg een kleurenpatroon.Verwijder het programma en tik volgende regels in :

```
10 SCREEN 2
20 LINE (10,10)-(100,100),,B
30 GOTO 30
```

Met dit programma hadden we de bedoeling om een vierkant te tekenen in de voorgrondkleur. Zoals uit het resultaat blijkt moeten we, om een echt vierkant te verkrijgen, corrigeren.
Wijzig regel 20 in :

```
20 LINE (10,10*1.4)-(100,100*1.4),,B
```

Door de Y-coordinaten met een factor 1,4 te vermenigvuldigen bekomen we een goed resultaat.

3.3. CIRCLE.

Met deze opdracht gaan we ellipsen, cirkels, cirkelbogen en taartpunten tekenen. Hiervoor hebben we coordinaten nodig voor het middelpunt, een straal, een kleur, een beginhoek, een eindhoek en een aanzichtsverhouding. Al deze gegevens, behalve middelpunt en straal kunnen weggelaten worden.
Tik het volgende programma in :

```
10 SCREEN 2
20 PI=3.1416:X=256/2:Y=192/2
30 CIRCLE (X,Y),50
1000 GOTO 1000
```

Dit geeft op het scherm een ellips in de voorgrondkleur met aanzichtsverhouding 1 omdat we deze niet ingevuld hebben.
Wijzig regel 30 in :

```
30 CIRCLE (X,Y),50,5,,,1.4
```

Dit geeft ons een cirkel in het blauw, door het invullen van de aanzichtsverhouding en de kleur. Het weglaten van de gegevens tussen de komma's geeft aan dat we hier niets wensen in te vullen.
Wijzig regel 30 in :

```
30 CIRCLE (X,Y),50,5,PI/2,PI,1.4
```

We verkrijgen nu een cikelboog door het invullen van het begin en eindpunt van de cirkel. De eenheid van deze gegevens is de radiaal en varieren van 0 tot 2PI. Door deze getallen negatief voor te stellen verkrijgen we taartpunten.
Wijzig regel 30 en voeg regel 40 bij :

```
30 CIRCLE (X,Y),50,5,-PI/2,-PI,1.4
40 CIRCLE (X+8,Y+8*1.4),50,10,-PI,
   -PI/2,1.4
```

We verkrijgen nu een gele taartvorm waaruit een blauw stuk is gesneden.

3.4. PAINT.

Met deze opdracht gaan we een vlak, dat door een aaneengesloten lijn bepaald is, kleuren. Hiervoor hebben we een punt van dat vlak nodig, een vulkleur en een randkleur. Met SCREEN 2 is het niet mogelijk om de randkleur te laten verschillen van de vulkleur.Voeg de volgende regels bij het vorige programma :

```
50 PAINT (X-1,Y-1),5,5
60 PAINT (X+20,Y+20),10,10
```

Dit vult de taartpunten, die we in 3.3. hebben gemaakt, met hun respectievelijke kleur.

3.5. POINT.

Terwijl de vorige opdrachten figuren 'op' het scherm bracht, gaan we met deze functie informatie 'van' het scherm halen. We kunnen de kleur van een bepaald punt aan een variable toekennen of, de coordinaten opzoeken van een punt met een bepaalde kleur.

3.6. DRAW.

In tegenstelling met de vorige grafishe opdrachten gebruikt deze opdracht geen vast aantal parameters, maar zijn ze afhankelijk van wat de gebruiker in een rijvariable plaatst.
Tik het volgende programma in :

```
10 SCREEN 2
20 DRAW "BM256/2,192"
30 FOR I=4 TO 240 STEP 12
40 DRAW "S=I;BURD2L2U2RBD"
50 NEXT I
60 GOTO 60
```

Met régel 10 schakelen we over op het grafisch scherm. Met regel 20 begeven we ons naar het midden van het scherm zonder een lijn te trekken, we doen dit door voor de opdracht de letter B te zetten, dit als bedoeling 'Blind'. De letter M wil zeggen 'Move', in het nederlands 'beweeg'. Met de FOR-NEXT lus gaan we telkens een andere schaalfactor instellen als de lus doorlopen wordt. Deze factor brengen we binnen onze rijvariable door S (de schaalfactor) er aan gelijk te maken. We moeten zulk een aktie wel afsluiten met een commapunt.
In regel dertig gebeurt dit door 'S=I;'.
'BU' zorgt ervoor dat we vanuit het middelpunt een eenheid omhoog gaan, U=Up=omhoog, zonder een lijn te trekken.
'RD2L2U2R' zorgt ervoor dat er een zichtbaar vierkant wordt getekend door vervolgens een eenheid naar rechts, R=rechts, twee eenheden naar beneden, D=Down=beneden, twee eenheden naar links L=links, twee eenheden naar omhoog, en een eenheid naar rechts te gaan zodat we terug op het vertrekpunt zijn.
'BD' zorgt ervoor dat we terug in het middelpunt terecht komen. Bij de volgende doorgangen van de lus is de schaalfactor vergroot zodat we een reeks vierkanten tekenen.
Verder kunnen we met 'C' de kleur veranderen, met 'A' het hele assenstelsel draaien, met 'H', 'G', 'F' en 'E' lijnen trekken onder een hoek van 45 graden en met 'N' als voorvoegsel geen nieuwe coordinaten laten berekenen.

3.7. Tekst in een grafish scherm.

We kunnen op een eenvoudige manier tekst in een grafish scherm zetten door eenvoudig een data bestand te openen, niet voor de datarecorder maar voor het grafisch scherm.
Tik bij het programma dat we in 3.4. hadden volgende regels bij in :

```
70 OPEN "GRP:" FOR OUTPUT AS #1
80 PRESET (65,0)
90 PRINT #1,"Totale autokosten"
100 PRESET (40,65)
110 PRINT #1,"Benzine"
120 PRESET (170,130)
130 PRINT #1,"Aankoop"
```

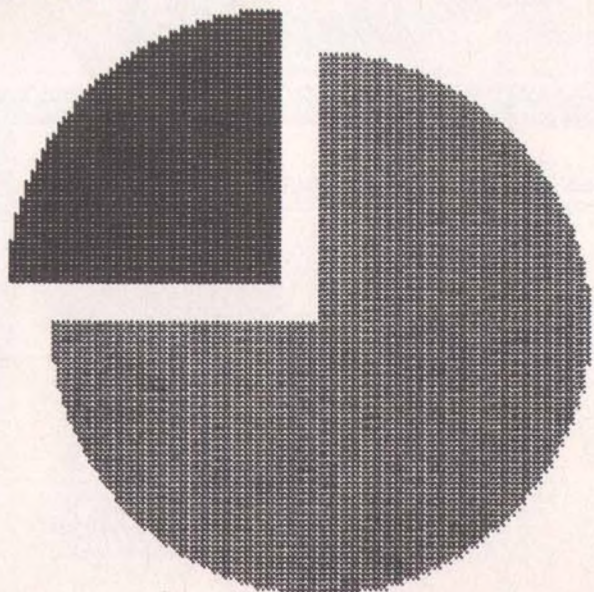
Met regel 70 openen we het bestand naar het grafisch scherm. Met regel 80, 100 en 120 bepalen we telkens de plaats waar de tekst moet beginnen. Met regel 90, 110 en 130 drukken we telkens onze boodschap af.

C.F.

```
1 COLOR 1,15,15
10 SCREEN 2
20 PI=3.1416:X=256/2:Y=192/2
30 CIRCLE (X,Y),50,5,-PI/2,-PI,1
   .4
40 CIRCLE (X+8,Y+8),50,10,-PI,-P
   I/2,1.4
50 PAINT (X-1,Y-1),5,5
60 PAINT (X+20,Y+20),10,10
70 OPEN "GRP:" FOR OUTPUT AS #1
80 PRESET (65,0)
90 PRINT #1,"Totale autokosten"
100 PRESET (40,65)
110 PRINT #1,"Benzine"
120 PRESET (170,130)
130 PRINT #1,"Aankoop"
1000 GOTO 1000
```

Totale autokosten

Benzine



Aankoop

NAME TABLE

0
255
256 1
511
512
767

PATTERN TABLE

0
2047
2048
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
4096
4097
6142

SCREEN

0
32
31
63
255
511
767

COLOUR TABLE

0
2047
2048
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
4096
4097
6142



Figuur 7

SVI

SPECTRA VIDEO
SV 728 MSX
SV 738 MSX



SV 738 X'PRESS

De eerste portable MSX computer, die bovendien voorzien is van het krachtige CP/M 2.2 operating system. De X'PRESS is één van de eerste MSX computers, die voorzien is van een ingebouwde disk drive $3^{1/2}$", 360K, die functioneert onder de drie operating systems nl. Disk Basic, MSX-DOS en CP/M.

De 80 kolommenkaart is reeds ingebouwd voor het draaien van professionele CP/M programma's. Tevens beschikt het apparaat over de mogelijkheid om ook MSX-Dos onder 40 en 80 kolommen te draaien.

Buiten de standaard MSX I/0 poorten en de standaard Centronics Interface, is ook de RS 232C interface ingebouwd voor datacommunicatie.

De X'PRESS is bovendien voorzien van een aansluiting voor een tweede disk drive voor een totale capaciteit van 720K, terwijl via de Spectravideo adapter SV 777 ook de $5^{1/4}$" disk drive kan worden aangesloten via het ROM slot om de formaten $5^{1/4}$" en $3^{1/2}$" heen en weer te kopieëren.

De SV 738 X'PRESS zal compleet worden geleverd inclusief draagtas, MSX-DOS diskette, CP/M 2.2 diskette en een aantal professionele gebruikssoftware.
Tevens wordt een utility programma meegeleverd waardoor de disk drive compatible zal zijn met de Bondwell 2 software, zoals Wordstar, Mail Merge, Calcstar etc.

SV 738 «X« press

- MSX
- CP/M 2.2
- Disc Basic
- MSX-DOS
- 32K ROM
- 80K RAM
- 16K video RAM
- Built-in drive $3^{1/2}$"
- Enlarged cursor control keys
- MSX-DOS 40 and 80 column
- CP/M 80 column
- Fullstroke keyboard
- Romslot I/0 port
- 2 Joystick and data cassette ports
- Centronics interface
- RS 232 interface
- I/0 port for second drive $3^{1/2}$" or $5^{1/4}$"
- Audio/video output
- VHF modulator built-in
- RGB optional
- Including carrying bag

Sofware included on disc

- MSX-DOS
- CP/M 2.2
- Demo
- Spread sheet
- Data writer
- Memo writer
- Report writer
- CP/M utility for MicroPro CP/M programs $3^{1/2}$"

SV 738 : 36.990 BTW in
SV 728 : 18.990 BTW in

Electronics Nederland bv
Tijnmuiden 15/19, Amsterdam
Tel (020) 139960, Telex 13406
Electronics Belgium NV
Brixtonlaan 1H, 1930 Zaventem
Tel (2) 7208945, Telex 67212

MSX